gAGE

gAGE/UPC research group of Astronomy and Geomatics
BarcelonaTECH, Spain

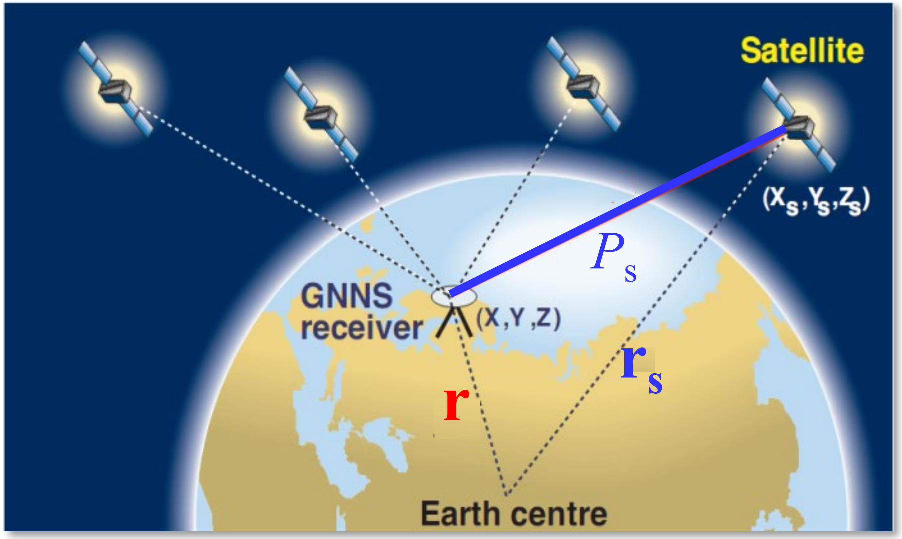
Lecture 8

Solving navigation equations

Professors: Dr. J. Sanz Subirana, Dr. J.M. Juan Zornoza
and Dr. Adrià Rovira García

Contents

Introduction: Linear model and Prefit-residuals



Input:

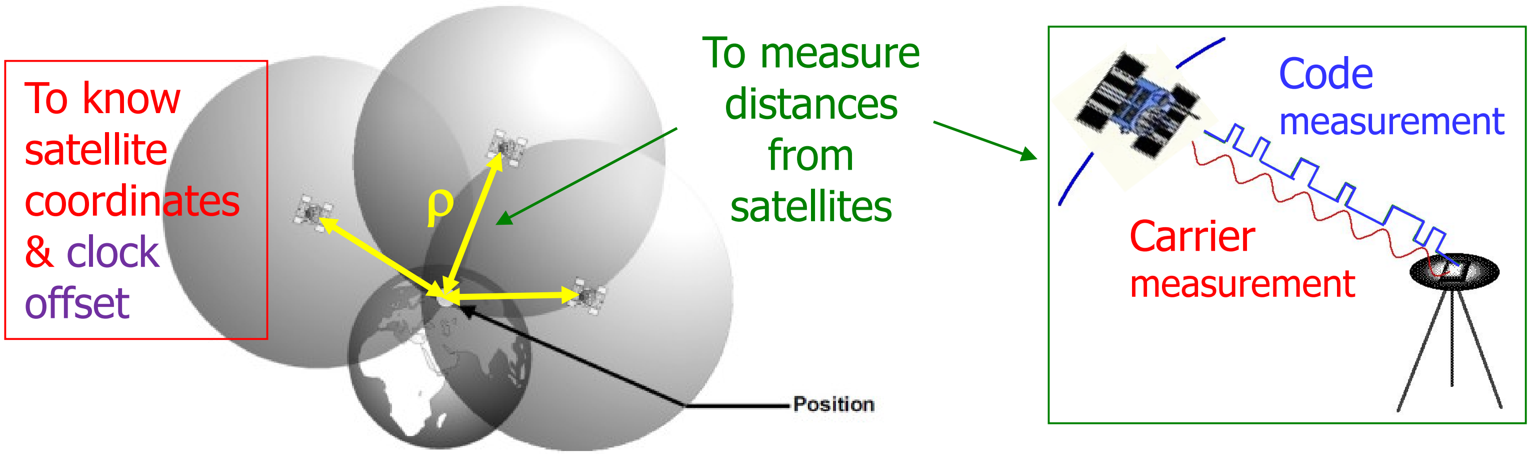
- **Pseudoranges (receiver-satellite j):** P_s
- **Navigation message. In particular:**
  - **Satellites position when transmitting signal:** $\mathbf{r}_s = (x_s, y_s, z_s)$
  - **Offsets of satellite clocks:** dt_s

  (satellites = 1, 2,…n)   (n>=4)

Unknowns:

- **Receiver position:** $\mathbf{r} = (x, y, z)$
- **Receiver clock offset:** dT

GNSS positioning concept



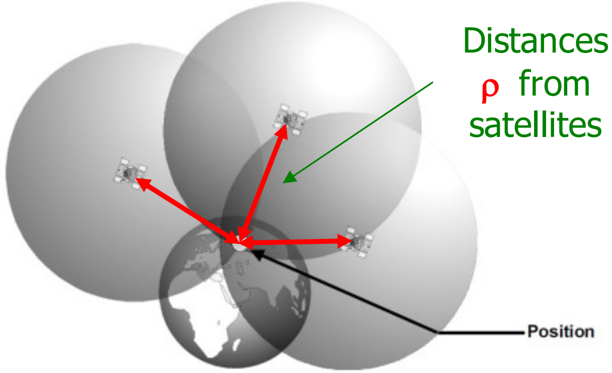
This picture is from https://gpsfleettrackingexpert.wordpress.com

- GNSS uses technique of "**triangulation**" to find user location
- To "**triangulate**" a GNSS receiver needs:
  - **To know the satellite coordinates** and clock synchronism errors:
    ➔ Satellites broadcast orbits parameters and clock offsets.
  - **To measure distances from satellites**:
    ➔ This is done measuring the **traveling time** of radio signals: ("Pseudo-ranges": **Code** and **Carrier** measurements)

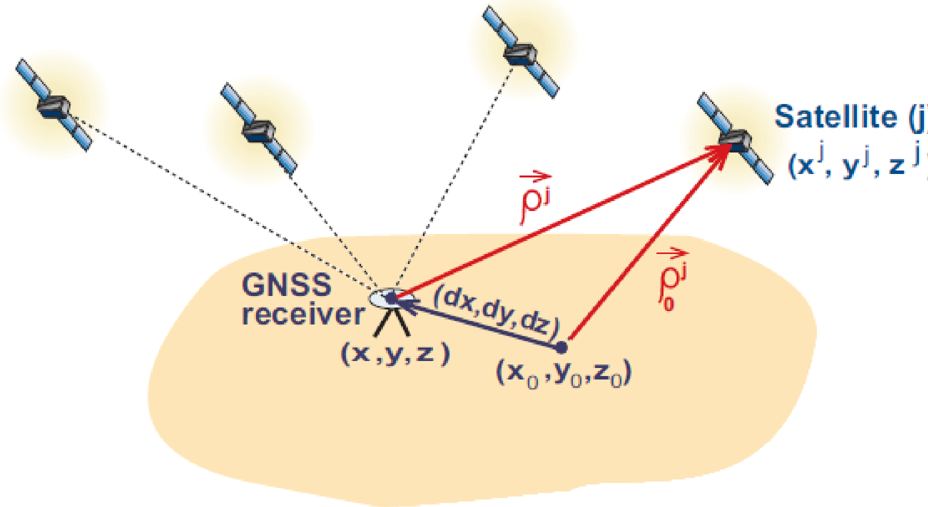
    ➔ Measurements must be corrected by several error sources:
    Atmospheric propagation, relativity, clock offsets, instrumental delays…

$$C1_{rec}^{sat} = \rho_{rec}^{sat} + c\cdot(dt_{rec} - dt^{sat}) + Trop_{rec}^{sat} + Ion_{1rec}^{sat} + K_{1rec} + TGD^{sat} + \varepsilon_1$$

Figure 6.1: Geometric concept of GNSS positioning: Equations



This picture is from https://gpsfleettrackingexpert.wordpress.com



Then, linearising the satellite–receiver geometric range

$$\rho^j(x,y,z) = \sqrt{(x^j - x)^2 + (y^j - y)^2 + (z^j - z)^2}$$

gives, for the approximate solution $\mathbf{r}_0 = (x_0, y_0, z_0)$,

$$\rho^j = \rho_0^j + \frac{x_0 - x^j}{\rho_0^j}dx + \frac{y_0 - y^j}{\rho_0^j}dy + \frac{z_0 - z^j}{\rho_0^j}dz$$

with $dx = x - x_0, \; dy = y - y_0, \; dz = z - z_0$

$$C1_{rec}^{sat}[\text{modelled}] = \rho_{rec,0}^{sat} - c\left(d\bar{t}^{\,sat} + \Delta rel^{sat}\right) + Trop_{rec}^{sat} + Ion_{1rec}^{sat} + TGD^{sat}$$

For each satellite in view

Iono+Tropo+TGD…

$$C1_{rec}^{sat} = \rho_{rec}^{sat} + c\cdot(dt_{rec} - dt^{sat}) + \sum \delta_k + \varepsilon$$

Linearising ρ around an 'a priori' receiver position $(x_{0,rec}, y_{0,rec}, z_{0,rec})$

$$= \rho_{0,rec}^{sat} + \frac{x_{0,rec} - x^{sat}}{\rho_{0,rec}^{sat}}\Delta x_{rec} + \frac{y_{0,rec} - y^{sat}}{\rho_{0,rec}^{sat}}\Delta y_{rec} + \frac{z_{0,rec} - z^{sat}}{\rho_{0,rec}^{sat}}\Delta z_{rec} + c\left(dt_{rec} - dt^{sat}\right) + \sum \delta_k$$

where:

$$\Delta x_{rec} = x_{rec} - x_{0,rec} \quad ; \quad \Delta y_{rec} = y_{rec} - y_{0,rec} \quad ; \quad \Delta z_{rec} = z_{rec} - z_{0,rec}$$

Prefit-residuals (Prefit)

$$C1_{rec}^{sat} - \left[\rho_{0,rec}^{sat} + c\,dt^{sat} - \sum \delta_k\right] = \frac{x_{0,rec} - x^{sat}}{\rho_{0,rec}^{sat}}\Delta x_{rec} + \frac{y_{0,rec} - y^{sat}}{\rho_{0,rec}^{sat}}\Delta y_{rec} + \frac{z_{0,rec} - z^{sat}}{\rho_{0,rec}^{sat}}\Delta z_{rec} + c\,dt_{rec}$$

measurement

computed

unknown

For all sat. in view

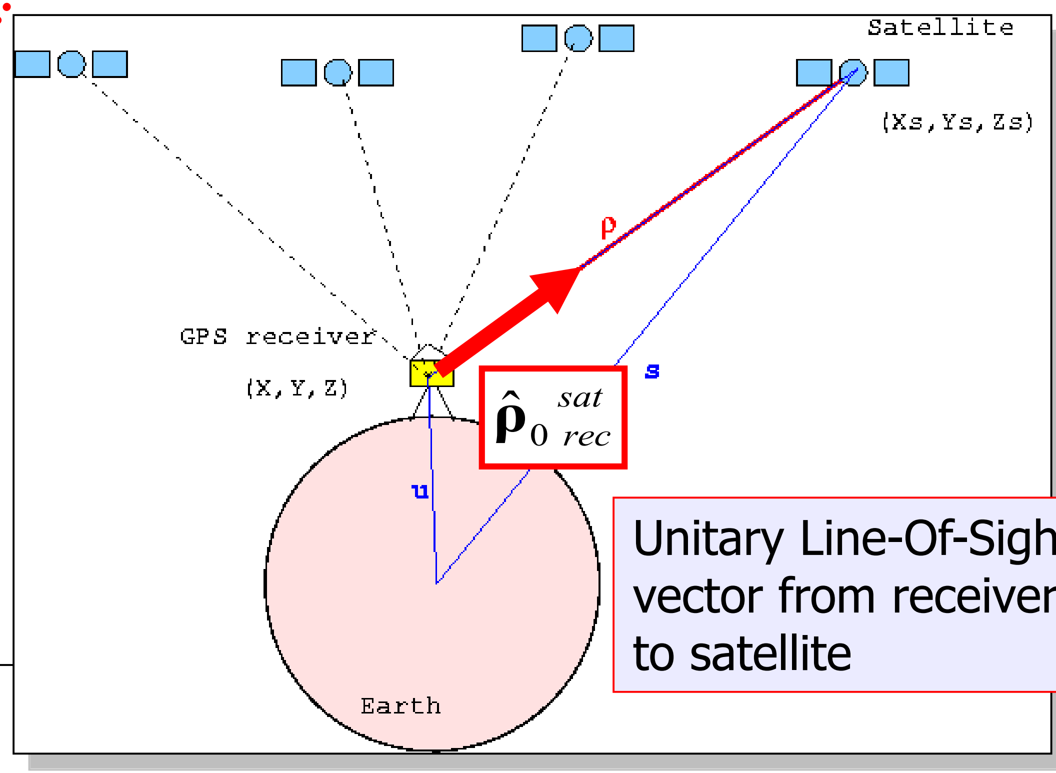
$$\begin{bmatrix} Prefit^1 \\ Prefit^2 \\ \cdots\cdots \\ Prefit^n \end{bmatrix} = \begin{bmatrix} \frac{x_{0,rec}-x^{sat1}}{\rho^{sat1}_{0,rec}} & \frac{y_{0,rec}-y^{sat1}}{\rho^{sat1}_{0,rec}} & \frac{z_{0,rec}-z^{sat1}}{\rho^{sat1}_{0,rec}} & 1 \\ \frac{x_{0,rec}-x^{sat2}}{\rho^{sat2}_{0,rec}} & \frac{y_{0,rec}-y^{sat2}}{\rho^{sat2}_{0,rec}} & \frac{z_{0,rec}-z^{sat2}}{\rho^{sat2}_{0,rec}} & 1 \\ & \cdots\cdots\cdots & & \\ \frac{x_{0,rec}-x^{sat\,n}}{\rho^{sat\,n}_{0,rec}} & \frac{y_{0,rec}-y^{sat\,n}}{\rho^{sat\,n}_{0,rec}} & \frac{z_{0,rec}-z^{sat\,n}}{\rho^{sat\,n}_{0,rec}} & 1 \end{bmatrix} \begin{bmatrix} \Delta x_{rec} \\ \Delta y_{rec} \\ \Delta z_{rec} \\ c\,dt_{rec} \end{bmatrix}$$

Observations (measured-computed)

Geometry of rays

$$-\frac{\boldsymbol{\rho}^{T\ sat\,n}_{0\ \ rec}}{\rho^{sat\,n}_{0\ rec}}$$

$$\hat{\boldsymbol{\rho}}^{T\ sat\,n}_{0\ rec} \equiv \frac{\boldsymbol{\rho}^{T\ sat\,n}_{0\ \ rec}}{\rho^{sat1}_{0\ rec}}$$



$\hat{\boldsymbol{\rho}}^{sat}_{0\ rec}$

Unitary Line-Of-Sight vector from receiver to satellite

(x,y,z) coordinates

$$\begin{bmatrix} Prefit^1 \\ Prefit^2 \\ \cdots\cdots \\ Prefit^n \end{bmatrix} = \begin{bmatrix} \frac{x_{0,rec}-x^{sat1}}{\rho_{0,rec}^{sat1}} & \frac{y_{0,rec}-y^{sat1}}{\rho_{0,rec}^{sat1}} & \frac{z_{0,rec}-z^{sat1}}{\rho_{0,rec}^{sat1}} & 1 \\ \frac{x_{0,rec}-x^{sat2}}{\rho_{0,rec}^{sat2}} & \frac{y_{0,rec}-y^{sat2}}{\rho_{0,rec}^{sat2}} & \frac{z_{0,rec}-z^{sat2}}{\rho_{0,rec}^{sat2}} & 1 \\ & \cdots\cdots\cdots & & \\ \frac{x_{0,rec}-x^{sat\,n}}{\rho_{0,rec}^{sat\,n}} & \frac{y_{0,rec}-y^{sat\,n}}{\rho_{0,rec}^{sat\,n}} & \frac{z_{0,rec}-z^{sat\,n}}{\rho_{0,rec}^{sat\,n}} & 1 \end{bmatrix} \begin{bmatrix} \Delta x_{rec} \\ \Delta y_{rec} \\ \Delta z_{rec} \\ c\,dt_{rec} \end{bmatrix}$$

Observations
(measured-computed)

Geometry of rays

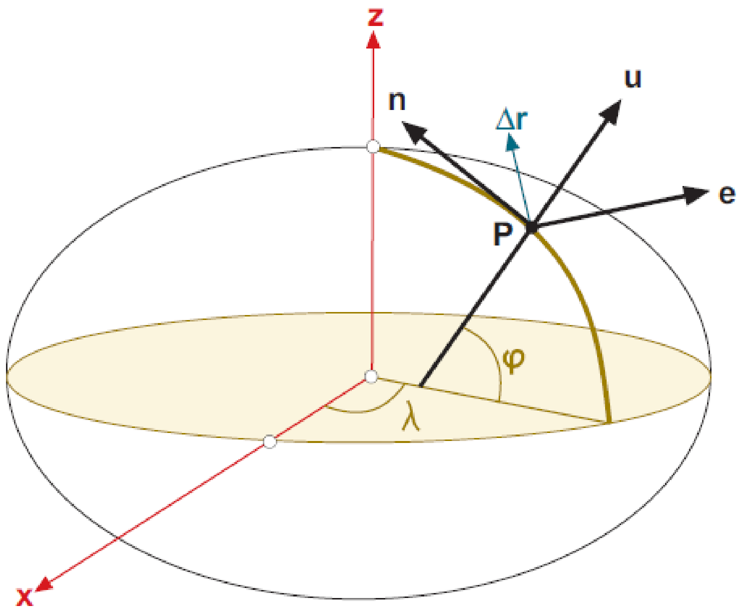
$$-\hat{\boldsymbol{\rho}}_{0\ rec}^{T\ sat\,n}$$

$$\begin{bmatrix} Prefit^1 \\ Prefit^2 \\ \cdots\cdots \\ Prefit^n \end{bmatrix} = \begin{bmatrix} -\hat{\boldsymbol{\rho}}_{0\ rec}^{T\ sat1} & 1 \\ -\hat{\boldsymbol{\rho}}_{0\ rec}^{T\ sat2} & 1 \\ \cdots & \cdots \\ -\hat{\boldsymbol{\rho}}_{0\ rec}^{T\ sat\,n} & 1 \end{bmatrix} \begin{bmatrix} \Delta \mathbf{r}_{rec} \\ c\,dt_{rec} \end{bmatrix}$$

(e,n,u) coordinates

$$\begin{bmatrix} Prefit^1 \\ Prefit^2 \\ \cdots\cdots \\ Prefit^n \end{bmatrix} = \begin{bmatrix} -\cos el^1 \sin az^1 & -\cos el^1 \cos az^1 & -\sin el^1 & 1 \\ -\cos el^2 \sin az^2 & -\cos el^2 \cos az^2 & -\sin el^2 & 1 \\ & \cdots\cdots\cdots & & \\ -\cos el^n \sin az^n & -\cos el^n \cos az^n & -\sin el^n & 1 \end{bmatrix} \begin{bmatrix} \Delta e_{rec} \\ \Delta n_{rec} \\ \Delta u_{rec} \\ c\,dt_{rec} \end{bmatrix}$$

From ECEF (x,y,z) to Local (e,n,u) coordinates



$$\begin{bmatrix} \Delta e \\ \Delta n \\ \Delta u \end{bmatrix} = \mathbf{R}_1[\pi/2-\varphi]\,\mathbf{R}_3[\pi/2+\lambda] \begin{bmatrix} \Delta x \\ \Delta y \\ \Delta z \end{bmatrix}$$

$$\hat{\mathbf{e}} = (-\sin\lambda\,,\ \cos\lambda\,,\ 0)$$
$$\hat{\mathbf{n}} = (-\cos\lambda\sin\varphi\,,\ -\sin\lambda\sin\varphi\,,\ \cos\varphi)$$
$$\hat{\mathbf{u}} = (\cos\lambda\cos\varphi\,,\ \sin\lambda\ \cos\varphi\,,\ \sin\varphi)$$

$$\begin{bmatrix} \Delta e \\ \Delta n \\ \Delta u \end{bmatrix} = \begin{bmatrix} -\sin\lambda & \cos\lambda & 0 \\ -\cos\lambda\sin\varphi & -\sin\lambda\sin\varphi & \cos\varphi \\ \cos\lambda\cos\varphi & \sin\lambda\cos\varphi & \sin\varphi \end{bmatrix} \begin{bmatrix} \Delta x \\ \Delta y \\ \Delta z \end{bmatrix}$$

COMMENTS:

$$\begin{bmatrix} Prefit^1 \\ Prefit^2 \\ \cdots\cdots \\ Prefit^n \end{bmatrix} = \begin{bmatrix} \frac{x_{0,rec}-x^{sat1}}{\rho_{0,rec}^{sat1}} & \frac{y_{0,rec}-y^{sat1}}{\rho_{0,rec}^{sat1}} & \frac{z_{0,rec}-z^{sat1}}{\rho_{0,rec}^{sat1}} & 1 \\ \frac{x_{0,rec}-x^{sat2}}{\rho_{0,rec}^{sat2}} & \frac{y_{0,rec}-y^{sat2}}{\rho_{0,rec}^{sat2}} & \frac{z_{0,rec}-z^{sat2}}{\rho_{0,rec}^{sat2}} & 1 \\ & \cdots\cdots & & \\ \frac{x_{0,rec}-x^{sat\,n}}{\rho_{0,rec}^{sat\,n}} & \frac{y_{0,rec}-y^{sat\,n}}{\rho_{0,rec}^{sat\,n}} & \frac{z_{0,rec}-z^{sat\,n}}{\rho_{0,rec}^{sat\,n}} & 1 \end{bmatrix} \begin{bmatrix} \Delta x_{rec} \\ \Delta y_{rec} \\ \Delta z_{rec} \\ c\,dt_{rec} \end{bmatrix}$$

Of course, receiver coordinates $(x_{rec}, y_{rec}, z_{rec})$ are not known (they are the target of this problem). But, we can always assume that an "approximate position $(x_{0,rec}, y_{0,rec}, z_{0,rec})$ is known".

Thence, the navigation problem will consist on:

1.- To start from an approximate value for receiver position $(x_{0,rec}, y_{0,rec}, z_{0,rec})$ e.g. the Earth's centre) to linearise the equations

2.- With the pseudorange measurements and the navigation equations, compute the correction $(\Delta x_{rec}, \Delta y_{rec}, \Delta z_{rec})$ to have improved estimates: $(x_{rec}, y_{rec}, z_{rec}) = (x_{0,rec}, y_{0,rec}, z_{0,rec}) + (\Delta x_{rec}, \Delta y_{rec}, \Delta z_{rec})$

3.- Linearise the equations again, about the new improved estimates, and iterate until the change in the solution estimates is sufficiently small.

For all sat in view

$$\begin{bmatrix} Prefit^1 \\ Prefit^2 \\ \cdots\cdots \\ Prefit^n \end{bmatrix} = \begin{bmatrix} \frac{x_{0,rec}-x^{sat1}}{\rho_{0,rec}^{sat1}} & \frac{y_{0,rec}-y^{sat1}}{\rho_{0,rec}^{sat1}} & \frac{z_{0,rec}-z^{sat1}}{\rho_{0,rec}^{sat1}} & 1 \\ \frac{x_{0,rec}-x^{sat2}}{\rho_{0,rec}^{sat2}} & \frac{y_{0,rec}-y^{sat2}}{\rho_{0,rec}^{sat2}} & \frac{z_{0,rec}-z^{sat2}}{\rho_{0,rec}^{sat2}} & 1 \\ & \cdots\cdots\cdots & & \\ \frac{x_{0,rec}-x^{sat\,n}}{\rho_{0,rec}^{sat\,n}} & \frac{y_{0,rec}-y^{sat\,n}}{\rho_{0,rec}^{sat\,n}} & \frac{z_{0,rec}-z^{sat\,n}}{\rho_{0,rec}^{sat\,n}} & 1 \end{bmatrix} \begin{bmatrix} \Delta x_{rec} \\ \Delta y_{rec} \\ \Delta z_{rec} \\ c\,dt_{rec} \end{bmatrix}$$

Observations
(measured-computed)

Geometry of rays

Thence, the basic linearized GPS measurement equation can be written as:

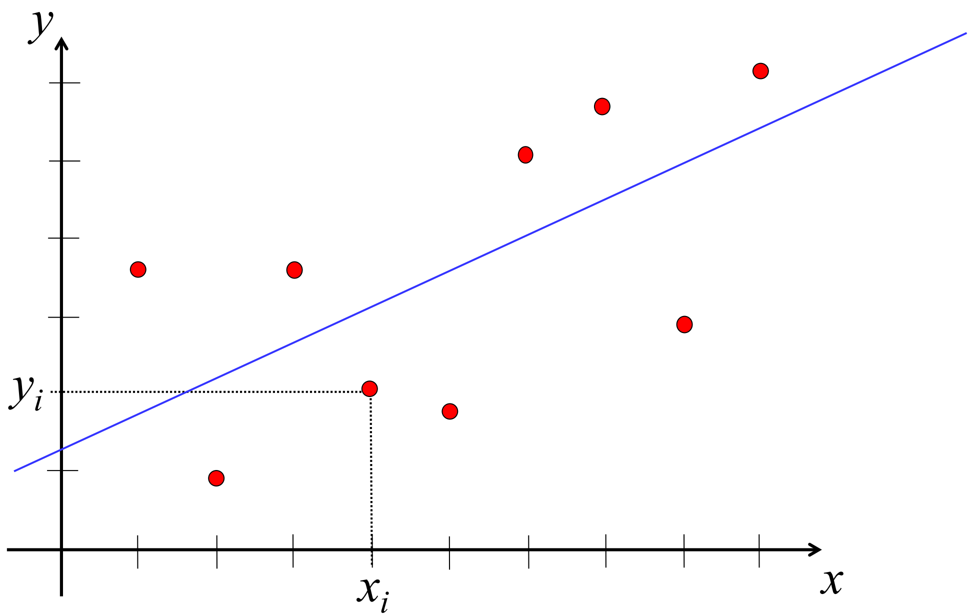
$$\mathbf{y} = \mathbf{G}\,\mathbf{x}$$

This is a linear system with, in general, $n \geq 4$ equations which we can solve using LS, WLS, Kalman filter,…

Contents

Least Squares solution (conceptual review)

As a driving problem, let us consider the problem of fitting a set of points (noisy measurements) to a straight line $y=mx+n$.



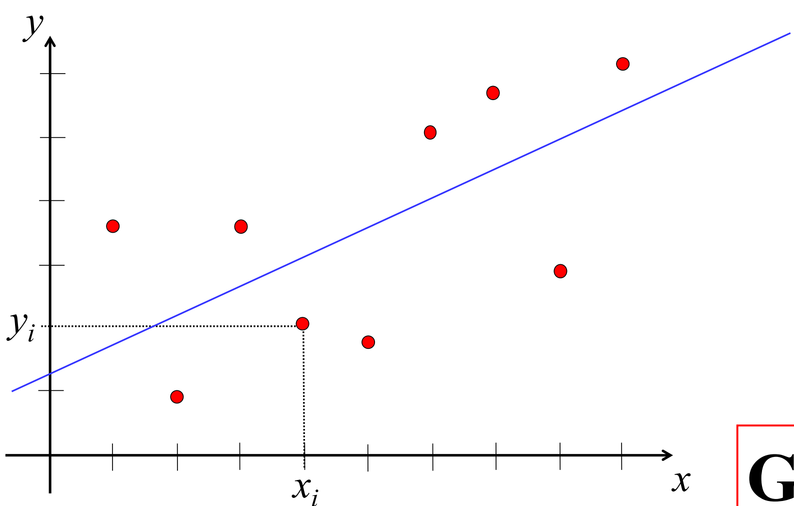
x	y
x_1	y_1
x_2	y_2
$\vdots$	$\vdots$
x_N	y_N

$$\begin{cases} y_1 \approx m\,x_1 + n \\ y_2 \approx m\,x_2 + n \\ \quad\vdots \\ y_N \approx m\,x_N + n \end{cases} \Rightarrow \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_N \end{bmatrix} = \begin{bmatrix} x_1 & 1 \\ x_2 & 1 \\ \vdots & \\ x_N & 1 \end{bmatrix} \begin{bmatrix} m \\ n \end{bmatrix} + \boldsymbol{\varepsilon} \Rightarrow \underset{N\times 1}{\mathbf{y}} = \underset{N\times 2}{\mathbf{G}}\,\underset{2\times 1}{\mathbf{p}} + \boldsymbol{\varepsilon}$$

$$\begin{cases} y_1 \approx m\,x_1 + n \\ y_2 \approx m\,x_2 + n \\ \vdots \\ y_N \approx m\,x_N + n \end{cases} \Rightarrow \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_N \end{bmatrix} = \begin{bmatrix} x_1 & 1 \\ x_2 & 1 \\ \vdots & \\ x_N & 1 \end{bmatrix} \begin{bmatrix} m \\ n \end{bmatrix} + \boldsymbol{\varepsilon} \Rightarrow \underset{N\times 1}{\mathbf{y}} = \underset{N\times 2}{\mathbf{G}}\,\underset{2\times 1}{\mathbf{p}} + \boldsymbol{\varepsilon}$$

This is an over-determined (**incompatible**) system of equations (due to the measurement noise $\boldsymbol{\varepsilon}$).

It is evident that there is no straight line passing over all the data points (red points). Thence, **we have to look for a solution that fits the measurements best in some sense.**



Note that, as $\mathbf{G}$ is not an squared matrix ($N>2$), we cannot try:

$$\mathbf{y} = \mathbf{G}\,\mathbf{p} \Rightarrow \mathbf{p} = \mathbf{G}^{-1}\mathbf{y}$$

But, $\mathbf{G}^T\mathbf{G}$ is a squared $(N \times N)$ matrix, thence, we can try:

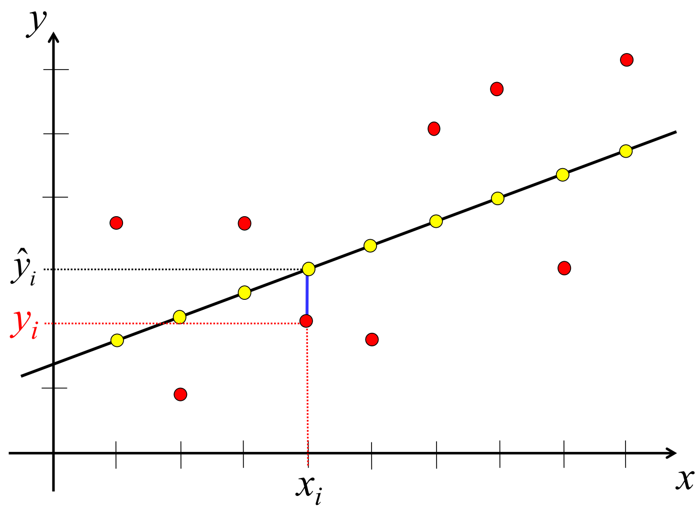
$$\mathbf{G}^T\mathbf{y} = \mathbf{G}^T\mathbf{G}\,\mathbf{p} \Rightarrow \hat{\mathbf{p}} = (\mathbf{G}^T\mathbf{G})^{-1}\mathbf{G}^T\mathbf{y}$$

Results from Linear Algebra:

1) $\exists\, (\mathbf{G}^T\mathbf{G})^{-1} \Leftrightarrow$ The columns of matrix G are linearly independents.

2) $\hat{\mathbf{p}} = (\mathbf{G}^T\mathbf{G})^{-1}\mathbf{G}^T\mathbf{y} \Leftrightarrow \min\|\mathbf{y}-\hat{\mathbf{y}}\| = \min\left(\sum_{i=1}^{N}(y_i-\hat{y}_i)^2\right)$ Least Squares solution

where $\hat{\mathbf{y}} = \mathbf{G}\,\hat{\mathbf{p}}$



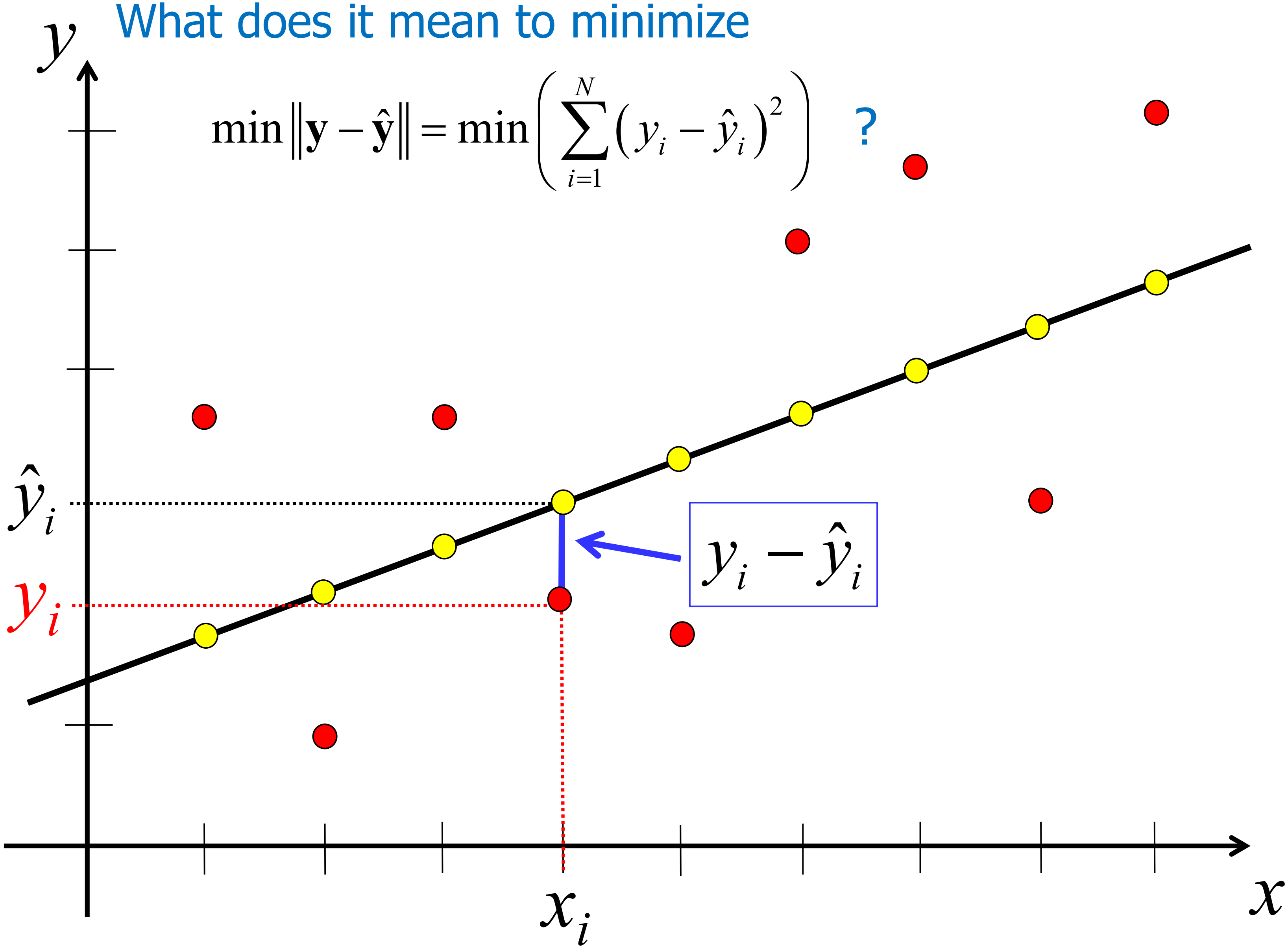
But, what is the physical meaning of the least square solution?

What does it mean the condition

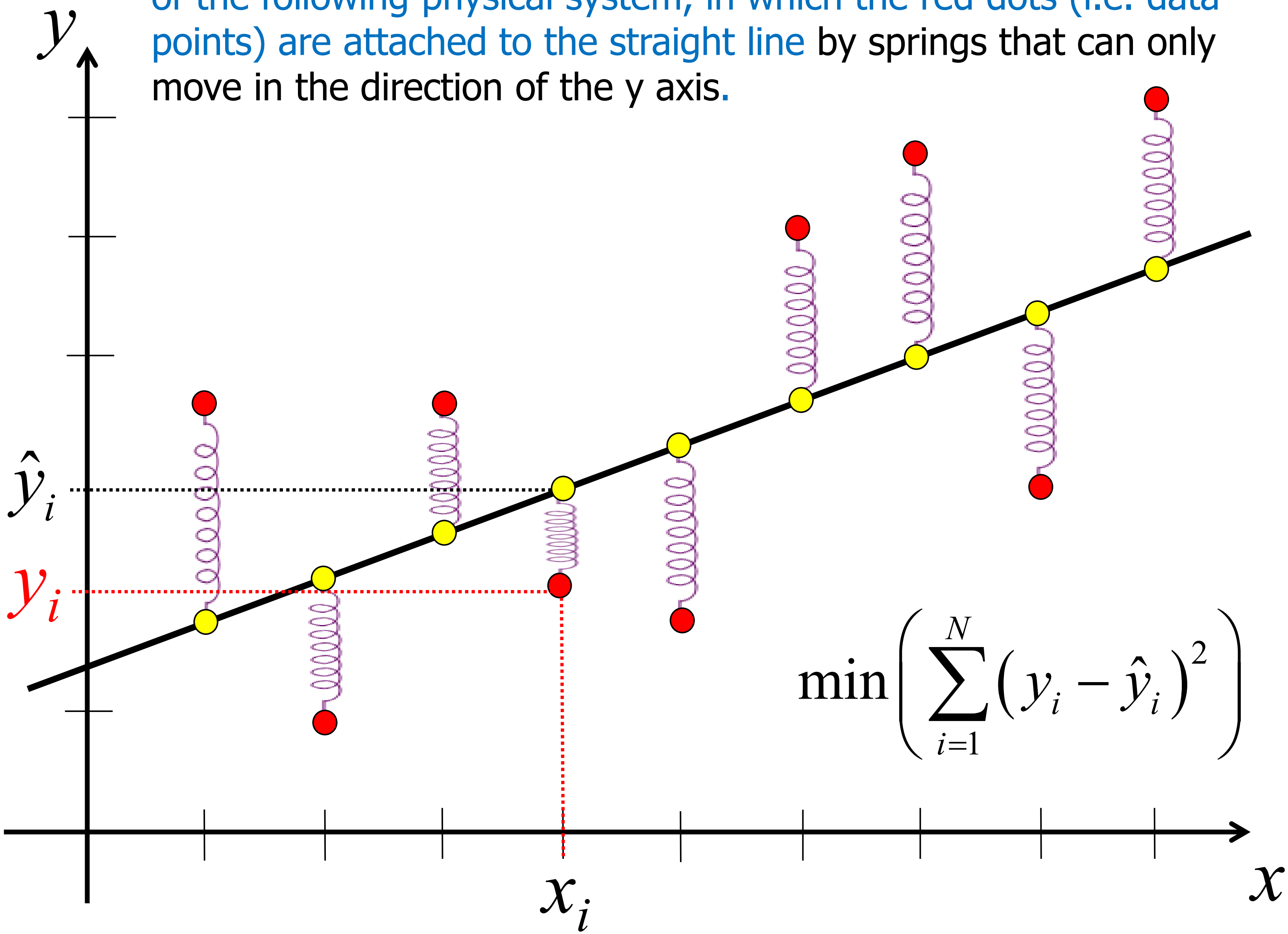
$$\min\|\mathbf{y}-\hat{\mathbf{y}}\| = \min\left(\sum_{i=1}^{N}(y_i-\hat{y}_i)^2\right) \quad ?$$

What is the physical meaning of the least square solution? What does it mean to minimize

$$\min \|\mathbf{y} - \hat{\mathbf{y}}\| = \min \left(\sum_{i=1}^{N} (y_i - \hat{y}_i)^2 \right) \quad ?$$



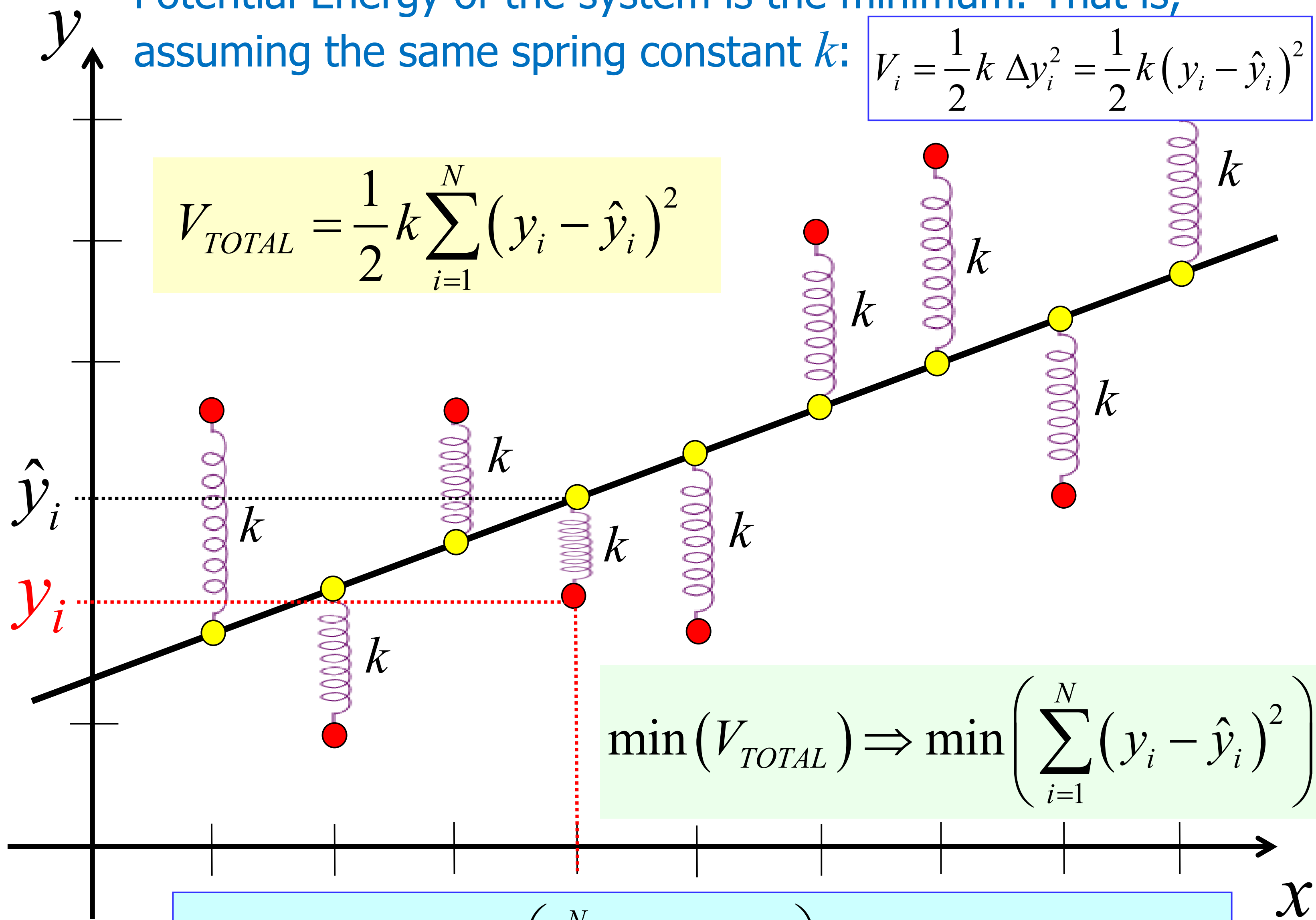
The Least Squares solution gives the solution of equilibrium of the following physical system, in which the red dots (i.e. data points) are attached to the straight line by springs that can only move in the direction of the y axis.



$$\min\left(\sum_{i=1}^{N}\left(y_i-\hat{y}_i\right)^2\right)$$

Indeed, the equilibrium solution is reached when the Total Potential Energy of the system is the minimum. That is, assuming the same spring constant k:

$$V_i = \frac{1}{2} k \, \Delta y_i^2 = \frac{1}{2} k \left(y_i - \hat{y}_i \right)^2$$



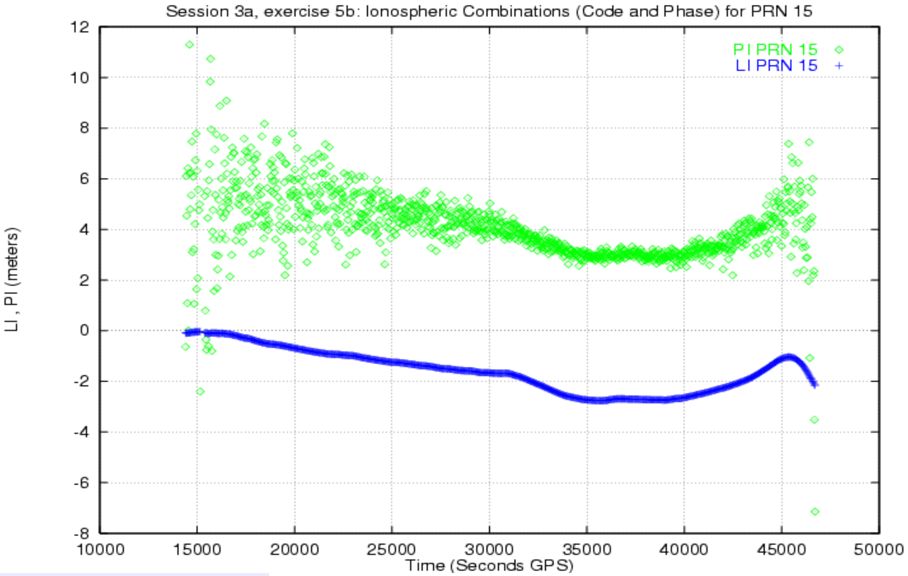
$$V_{TOTAL} = \frac{1}{2} k \sum_{i=1}^{N} \left(y_i - \hat{y}_i \right)^2$$

$$\min\left(V_{TOTAL}\right) \Rightarrow \min\left(\sum_{i=1}^{N} \left(y_i - \hat{y}_i \right)^2 \right)$$

$$\min \left\| \mathbf{y} - \hat{\mathbf{y}} \right\| = \min\left(\sum_{i=1}^{N} \left(y_i - \hat{y}_i \right)^2 \right) \Leftrightarrow \hat{\mathbf{p}} = (\mathbf{G}^T\mathbf{G})^{-1}\mathbf{G}^T\mathbf{y}$$

Let be the basic linearized G

$$\mathbf{y} = \mathbf{G}\,\mathbf{x}$$



- Least Squares solution:

$$\hat{\mathbf{x}} = \left(\mathbf{G}^t\mathbf{G}\right)^{-1}\mathbf{G}^t\mathbf{y}$$

The **same error** is assumed in all measurements

- Weighted Least Squares solution

If the measurements have **different errors**, the equations can be weighted by matrix $\mathbf{W}$:

$$\mathbf{W} = \begin{bmatrix} w_{y_1} & & 0 \\ & \ddots & \\ 0 & & w_{y_n} \end{bmatrix}$$

Uncorrelated errors are assumed

And the weighted least squares solution is:

$$\hat{\mathbf{x}} = \left(\mathbf{G}^t\mathbf{W}\,\mathbf{G}\right)^{-1}\mathbf{G}^t\;\mathbf{W}\mathbf{y}$$

$$\min \|\mathbf{y}-\hat{\mathbf{y}}\|^2_{\mathbf{W}} = \min\left[\sum_i w_i\left(y_i-\hat{y}_i\right)^2\right]$$

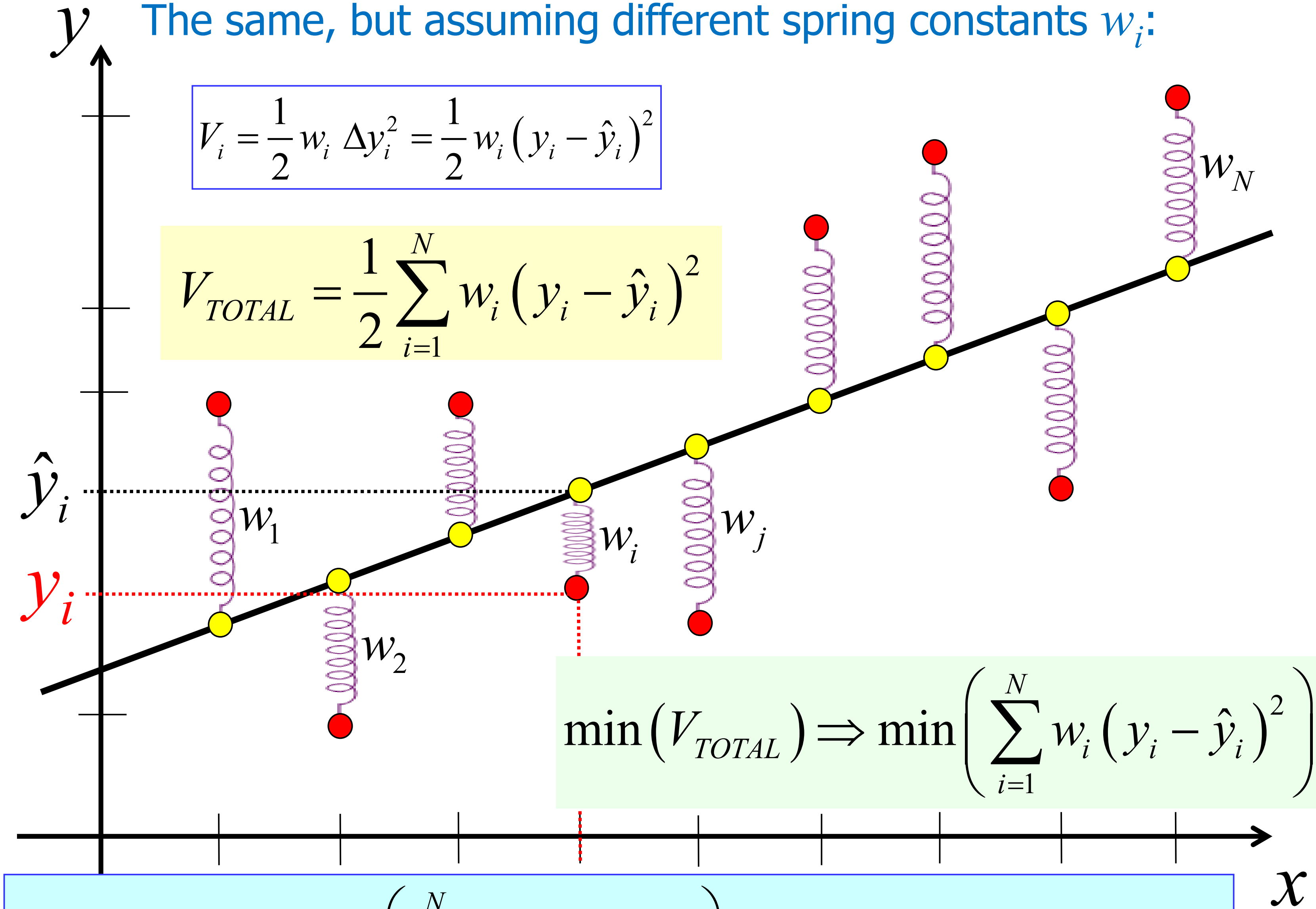
$$\hat{\mathbf{y}} = \mathbf{G}\,\hat{\mathbf{x}}$$

Weighted Least Squares solution:

The same, but assuming different spring constants w_i:

$$V_i = \frac{1}{2} w_i \, \Delta y_i^2 = \frac{1}{2} w_i \left(y_i - \hat{y}_i\right)^2$$

$$V_{TOTAL} = \frac{1}{2} \sum_{i=1}^{N} w_i \left(y_i - \hat{y}_i\right)^2$$



$$\min\left(V_{TOTAL}\right) \Rightarrow \min\left(\sum_{i=1}^{N} w_i \left(y_i - \hat{y}_i\right)^2\right)$$

$$\min \left\| \mathbf{y} - \hat{\mathbf{y}} \right\|_W = \min\left(\sum_{i=1}^{N} w_i \left(y_i - \hat{y}_i\right)^2\right) \Leftrightarrow \hat{\mathbf{p}} = (\mathbf{G}^T \mathbf{W} \mathbf{G})^{-1} \mathbf{G}^T \mathbf{W} \mathbf{y}$$

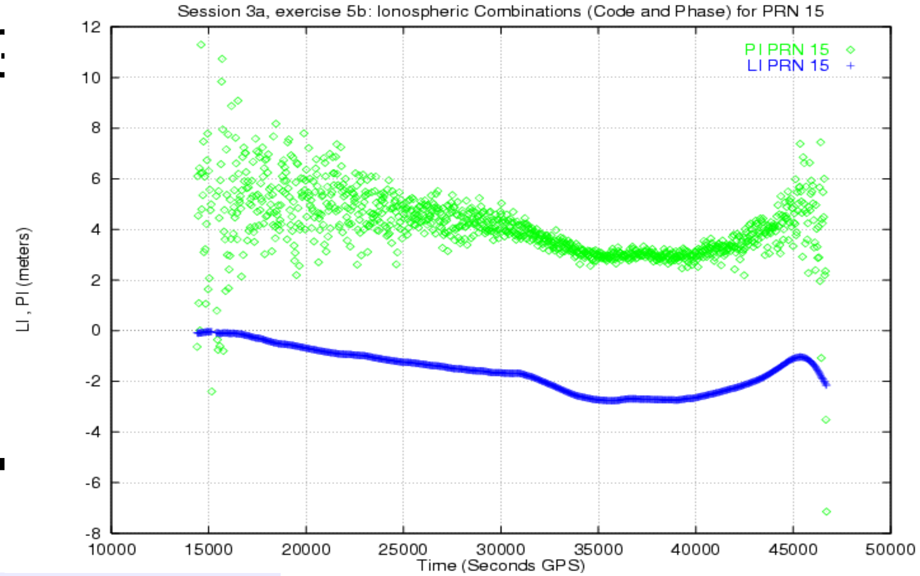
Contents

gAGE/UPC research group of Astronomy and Geomatics
BarcelonaTECH, Spain

gAGE

Let be the basic linearized G

$$\mathbf{y} = \mathbf{G}\,\mathbf{x}$$



- Least Squares solution:

$$\hat{\mathbf{x}} = \left(\mathbf{G}^t\mathbf{G}\right)^{-1}\mathbf{G}^t\mathbf{y}$$

The **same error** is assumed in all measurements

- Weighted Least Squares solution

If the measurements have **different errors**, the equations can be weighted by matrix $\mathbf{W}$:

$$\mathbf{W} = \begin{bmatrix} w_{y_1} & & 0 \\ & \ddots & \\ 0 & & w_{y_n} \end{bmatrix}$$

Uncorrelated errors are assumed

And the weighted least squares solution is:

$$\hat{\mathbf{x}} = \left(\mathbf{G}^t\mathbf{W}\,\mathbf{G}\right)^{-1}\mathbf{G}^t\,\mathbf{W}\mathbf{y}$$

$$\min \|\mathbf{y} - \hat{\mathbf{y}}\|^2_{\mathbf{W}} = \min\left[\sum_i w_i\left(y_i - \hat{y}_i\right)^2\right]$$

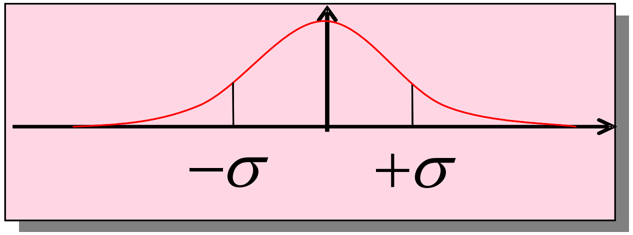
$$\hat{\mathbf{y}} = \mathbf{G}\,\hat{\mathbf{x}}$$

Assuming that measurements $\mathbf{Y}$ have **random errors with zero mean and variance σ^2**, and assuming that error sources for each satellite are **uncorrelated** with error sources for any other satellite, the following weighted matrix may be used:

$$\mathbf{W} = \begin{bmatrix} 1/\sigma_{y_1}^2 & & 0 \\ & \ddots & \\ 0 & & 1/\sigma_{y_n}^2 \end{bmatrix}$$

$$w_i = \frac{1}{\sigma_{y_i}^2} \rightarrow \sigma_{y_i}^2 \uparrow \quad \Rightarrow \quad w_i \downarrow$$

greater error → less weight



Best Linear Unbiased **Minimum Variance Estimator** (**BLUE**):

Let be "$\mathbf{P_y}$" the error covariance matrix for measurements $\mathbf{y}$.

If the weighting matrix is taken as $\mathbf{W} = \mathbf{P_y}^{-1}$, thence the Minimum Variance Solution is found:

$$\hat{x} = \left(\mathbf{G}^t \mathbf{P_y}^{-1} \mathbf{G}\right)^{-1} \mathbf{G}^t \, \mathbf{P_y}^{-1} \, \mathbf{y}$$

And the error covariance matrix for the estimation X is:

$$\mathbf{P_{\hat{x}}} = \left(\mathbf{G}^t \mathbf{P_y}^{-1} \mathbf{G}\right)^{-1}$$

7. Navigation equations system and LS solution (XYZ)

Repeat the previous exercise, but writing the system and computing the solution in (XYZ) coordinates. Also, compute GDOP, Precision Dilution Of Precision (PDOP) and TDOP.

Complete the following steps:

(a) The matrix **G** is now

$$\mathbf{G}_i = \left[\frac{x_0 - x^i}{\rho_0^i}, \frac{y_0 - y^i}{\rho_0^i}, \frac{z_0 - z^i}{\rho_0^i}, 1 \right]$$

where $\mathbf{r}_0 = (x_0, y_0, y_0)$ is the 'a priori' receiver coordinates at reception time, $\mathbf{r}^i = (x^i, y^i, y^i)$ are the satellite coordinates at transmission time, and $\rho_0^i = ||\mathbf{r}^i - \mathbf{r}_0||$.

Hint: Matrix **G** and prefit residual vector **y** can be generated directly from the `gLAB.out` output file as follows:[73]

```
grep MODEL gLAB.out | grep C1 | gawk 'BEGIN{x=4789032.6277;
     y=176595.0498;z=4195013.2503} {if ($4==300 && $6!=21)
     {r1=x-$11;r2=y-$12;r3=z-$13;r=sqrt(r1*r1+r2*r2+r3*r3);
     print $9-$10,r1/r,r2/r,r3/r,1}}' > M.dat
```

Vector **y** corresponds to the first column of file `M.dat` and matrix **G** to the last four columns.

See exercises 6 and 7, Session 5.2 in [RD-2]

The matrix **G** and vector **y** values computed by gLAB can be found by:

```
grep PREFIT gLAB.out | grep -v INFO |
 gawk '{if ($4==300 && $6!=21) print $8,$11,$12,$13,$14}'
```

(b) Compute the LS solution of the navigation system. Using Octave or MATLAB, upload the contents of file `M.dat` and execute the following instructions, as well:

```
y=M(:,1)
G=M(:,2:5)
x=inv(G'*G)*G'*y
```

The values computed by gLAB can be found by:

(X,Y,Z) coordinates:

```
grep OUTPUT gLAB.out | grep -v INFO |
              gawk '{if ($4==300) print $9,$10,$11}'
```

Receiver clock:

```
grep FILTER gLAB.out | grep -v INFO |
                     gawk '{if ($4==300) print $8}'
```

gAGE

gAGE/UPC research group of Astronomy and Geomatics
BarcelonaTECH, Spain

Contents

Kalman filtering:

It is based on computing the **weighted average between**:

- the measurement $\mathbf{y}(n)$ (i.e., at $t = t_n$)
- the prediction of the state $\hat{\mathbf{x}}^-(n)$ from previous estimation $\hat{\mathbf{x}}(n-1)$

1. Weighted average:

$$\begin{cases} \mathbf{y}(n) = \mathbf{G}(n)\mathbf{x}(n) \\ \hat{\mathbf{x}}^-(n) = \mathbf{x}(n) \end{cases}$$

Let's assume, that we have the prediction $\hat{\mathbf{x}}^-(n)$, with $\mathbf{P}_{\hat{\mathbf{x}}^-(n)}$ ce, it can be used to **add an additional set of equations** to the measurement equation $\mathbf{y} = \mathbf{G}\,\mathbf{x}$

$$\begin{bmatrix} \mathbf{y}(n) \\ \hat{\mathbf{x}}^-(n) \end{bmatrix} = \begin{pmatrix} \mathbf{G}(n) \\ \mathbf{I} \end{pmatrix} \mathbf{x}(n)$$

$$\mathbf{W} = \begin{pmatrix} \mathbf{R}_{\mathbf{y}(n)} & 0 \\ 0 & \mathbf{P}_{\hat{\mathbf{x}}^-(n)} \end{pmatrix}^{-1}$$

Kalman filtering:

It is based on computing the **weighted average** between:

- the measurement $\mathbf{y}(n),\ \mathbf{R}_{\mathbf{y}(n)}$ at $t = t_n$)
- the prediction $\hat{\mathbf{x}}^{-}(n),\ \mathbf{P}_{\hat{\mathbf{x}}^{-}(n)}$, from previous estimation $\hat{\mathbf{x}}(n-1),\ \mathbf{P}_{\hat{\mathbf{x}}(n-1)}$

1. Weighted average:

$$\begin{cases} \mathbf{y}(n) = \mathbf{G}(n)\mathbf{x}(n) \\ \hat{\mathbf{x}}^{-}(n) = \mathbf{x}(n) \end{cases}$$

Let's assume, that we have the prediction $\hat{\mathbf{x}}^{-}(n)$, with $\mathbf{P}_{\hat{\mathbf{x}}^{-}(n)}$ ce, it can be used to **add an additional set of equations** to the measurement equation **y= G x**

$$\begin{bmatrix} \mathbf{y}(n) \\ \hat{\mathbf{x}}^{-}(n) \end{bmatrix} = \begin{pmatrix} \mathbf{G}(n) \\ \mathbf{I} \end{pmatrix} \mathbf{x}(n)$$

$$\mathbf{W} = \begin{pmatrix} \mathbf{R}_{\mathbf{y}(n)} & 0 \\ 0 & \mathbf{P}_{\hat{\mathbf{x}}^{-}(n)} \end{pmatrix}^{-1}$$

$$\begin{bmatrix} \mathbf{y}(n) \\ \hat{\mathbf{x}}^-(n) \end{bmatrix} = \begin{pmatrix} \mathbf{G}(n) \\ \mathbf{I} \end{pmatrix} \mathbf{x}(n)$$

$$\mathbf{W} = \begin{pmatrix} \mathbf{R}_{\mathbf{y}(n)} & 0 \\ 0 & \mathbf{P}_{\hat{\mathbf{x}}^-(n)} \end{pmatrix}^{-1}$$

And the following solution of the previous equation system can be found with some elemental algebraic manipulations:

$$\hat{\mathbf{x}} = \left(\mathbf{G}^t \mathbf{P}_{\mathbf{y}}^{-1} \mathbf{G}\right)^{-1} \mathbf{G}^t \mathbf{P}_{\mathbf{y}}^{-1} \mathbf{y}$$

$$\mathbf{P}_{\hat{\mathbf{x}}} = \left(\mathbf{G}^t \mathbf{P}_{\mathbf{y}}^{-1} \mathbf{G}\right)^{-1}$$

$$\hat{\mathbf{x}}(n) = \mathbf{P}_{\hat{\mathbf{x}}(n)} \left[\mathbf{G}^t(n) \mathbf{R}_{\mathbf{y}(n)}^{-1} \mathbf{y}(n) + \mathbf{P}_{\hat{\mathbf{x}}^-(n)}^{-1} \hat{\mathbf{x}}^-(n) \right]$$

$$\mathbf{P}_{\hat{\mathbf{x}}(n)} = \left[\mathbf{G}^t(n) \mathbf{R}_{\mathbf{y}(n)}^{-1} \mathbf{G}(n) + \mathbf{P}_{\hat{\mathbf{x}}^-(n)}^{-1} \right]^{-1}$$

2.- Prediction

Scalar case:

Let's $\hat{x}_{n-1}$ be the state at epoch $n\text{-}1$ with variance $\sigma^2_{\hat{x}_{n-1}}$

The *simplest prediction model* is to assume that the prediction at epoch n is proportional to the state at epoch $n\text{-}1$. That is:

$$\hat{x}_n^- = \phi\, \hat{x}_{n-1}$$

Thence, existing a linear relation between $\hat{x}_{n-1}$ and $\hat{x}_n^-$, the variance of the prediction will be:

$$\sigma^2_{\hat{x}_n^-} = \phi^2\, \sigma^2_{\hat{x}_{n-1}} + q^2$$

An additional term is added to account for modeling error!

Generalization to the vector case:

$$\hat{x}_n^- = \phi \ \hat{x}_{n-1}$$
$$\sigma^2_{\hat{x}_n^-} = \phi^2 \ \sigma^2_{\hat{x}_{n-1}} + q^2$$

x_n	$\rightarrow$	$\mathbf{x}(n)$
ϕ	$\rightarrow$	$\mathbf{\Phi}(n)$
$\sigma^2_{x_n}$	$\rightarrow$	$\mathbf{P}_{\mathbf{x}(n)}$
q^2	$\rightarrow$	$\mathbf{Q}(n)$

$\mathbf{\Phi}(n)$: *transition matrix*
$\mathbf{Q}(n)$: *process noise matrix*

$$\hat{\mathbf{x}}^-(n) = \mathbf{\Phi}(n-1) \cdot \hat{\mathbf{x}}(n-1)$$
$$\mathbf{P}_{\hat{\mathbf{x}}^-(n)} = \mathbf{\Phi}(n-1) \cdot \mathbf{P}_{\hat{\mathbf{x}}(n-1)} \cdot \mathbf{\Phi}^t(n-1) + \mathbf{Q}(n-1)$$

Kalman filter (see kalman.f)

Measurements

$$\mathbf{y}(n) \qquad \mathbf{R}_{\mathbf{y}(n)}$$

$$\hat{\mathbf{x}}^-(n) \qquad \mathbf{P}_{\hat{\mathbf{x}}^-(n)}$$

Prediction

$$\hat{\mathbf{x}}^-(n) = \mathbf{\Phi}\,\hat{\mathbf{x}}(n-1)$$

$$\mathbf{P}_{\hat{\mathbf{x}}^-(n)} = \mathbf{\Phi}\,\mathbf{P}_{\hat{\mathbf{x}}(n-1)}\mathbf{\Phi}^t + \mathbf{Q}$$

Estimation

$$\hat{\mathbf{x}}(n) = \mathbf{P}_{\hat{\mathbf{x}}(n)} \cdot \left[\mathbf{G}^t\,\mathbf{R}^{-1}_{\mathbf{y}(n)}\,\mathbf{y}(n) + \mathbf{P}^{-1}_{\hat{\mathbf{x}}^-(n)}\hat{\mathbf{x}}^-(n)\right]$$

$$\mathbf{P}_{\hat{\mathbf{x}}(n)} = \left[\mathbf{G}^t\,\mathbf{R}^{-1}_{\mathbf{y}(n)}\,\mathbf{G} + \mathbf{P}^{-1}_{\hat{\mathbf{x}}^-(n)}\right]^{-1}$$

$$\hat{\mathbf{x}}(n) \qquad \mathbf{P}_{\hat{\mathbf{x}}(n)}$$

$$\hat{\mathbf{x}}(0) \qquad \mathbf{P}_{\hat{\mathbf{x}}(0)}$$

Initialization

Kalman filter (classical version)

Measurements

$$\mathbf{y}(n) \qquad \mathbf{R}_{\mathbf{y}(n)}$$

$$\hat{\mathbf{x}}^-(n) \qquad \mathbf{P}_{\hat{\mathbf{x}}^-(n)}$$

Estimation

$$\hat{\mathbf{x}}(n) = \hat{\mathbf{x}}^-(n) + \mathbf{K}(n)\cdot\left[\mathbf{y}(n) - \mathbf{G}(n)\cdot\hat{\mathbf{x}}^-(n)\right]$$

$$\mathbf{K}(n) = \mathbf{P}_{\hat{\mathbf{x}}^-(n)}\cdot\mathbf{G}^t(n)\cdot\left[\mathbf{G}(n)\cdot\mathbf{P}_{\hat{\mathbf{x}}^-(n)}\cdot\mathbf{G}^t(n) + \mathbf{R}_{\mathbf{y}(n)}\right]^{-1}$$

$$\mathbf{P}_{\hat{\mathbf{x}}(n)} = \left[\mathbf{I} - \mathbf{K}(n)\cdot\mathbf{G}(n)\right]\cdot\mathbf{P}_{\hat{\mathbf{x}}^-(n)}$$

Prediction

$$\hat{\mathbf{x}}^-(n) = \mathbf{\Phi}\,\hat{\mathbf{x}}(n-1)$$

$$\mathbf{P}_{\hat{\mathbf{x}}^-(n)} = \mathbf{\Phi}\,\mathbf{P}_{\hat{\mathbf{x}}(n-1)}\mathbf{\Phi}^t + \mathbf{Q}$$

$$\hat{\mathbf{x}}(0) \qquad \mathbf{P}_{\hat{\mathbf{x}}(0)}$$

Initialization

$$\hat{\mathbf{x}}(n) \qquad \mathbf{P}_{\hat{\mathbf{x}}(n)}$$

Contents

Some simple examples to define matrices Φ and Q

$$\hat{\mathbf{x}}^-(n) = \mathbf{\Phi}(n-1)\cdot\hat{\mathbf{x}}(n-1)$$

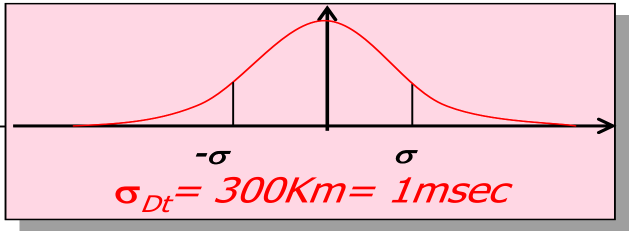
$$\mathbf{P}_{\hat{\mathbf{x}}^-(n)} = \Phi(n-1)\cdot P_{\hat{\mathbf{x}}(n-1)}\cdot\mathbf{\Phi}^t(n-1) + \mathbf{Q}(n-1)$$

a) Static positioning:

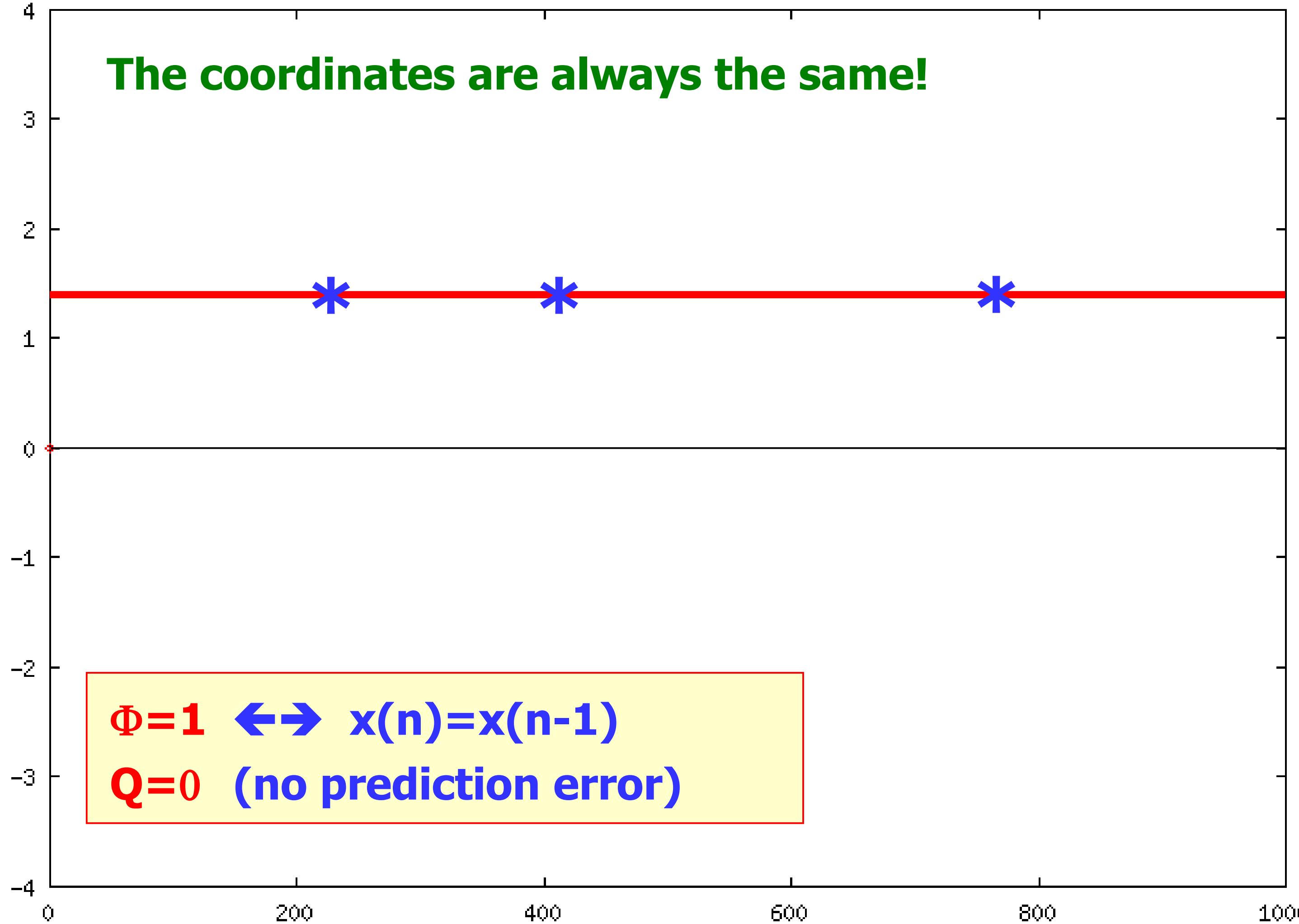
State vector to be determined is $X = (\Delta x_{rec}, \Delta y_{rec}, \Delta z_{rec}, Dt_{rec})$ where coordinates $(\Delta x_{rec}, \Delta y_{rec}, \Delta z_{rec})$ are considered **constant** (because receiver is fixed) and clock offset DT_{rec} is treated as **white noise** with zero mean and variance σ^2_{Dt} . In these conditions, matrices have the form:

$$\mathbf{\Phi}(n) = \begin{pmatrix} 1 & & & \\ & 1 & & \\ & & 1 & \\ & & & 0 \end{pmatrix} \qquad \mathbf{Q}(n) = \begin{pmatrix} 0 & & & \\ & 0 & & \\ & & 0 & \\ & & & \sigma^2_{DT} \end{pmatrix}$$

Being σ^2_{DT} process noise associated to clock offset (in some way, the uncertainty in clock value).



Constant

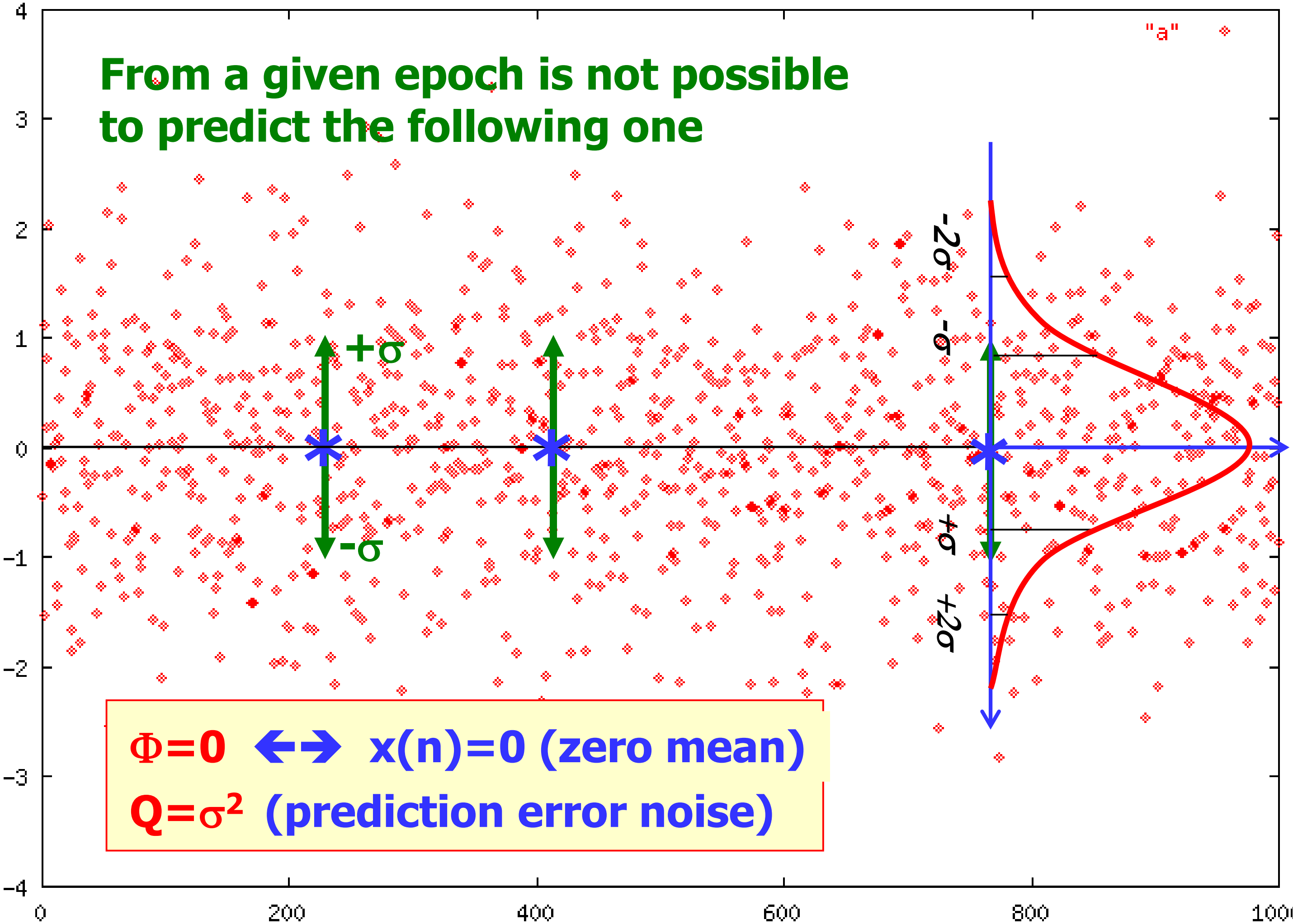


We can assure that, the next $\mathbf{x}(n)$ will be **the same as** $\mathbf{x}(n-1)$**.**

$$\hat{\mathbf{x}}^-(n) = \mathbf{\Phi(n-1)} \cdot \hat{\mathbf{x}}(n-1)$$

$$\mathbf{P}_{\hat{\mathbf{x}}^-(n)} = \mathbf{\Phi(n-1)} \cdot \mathbf{P}_{\hat{\mathbf{x}}(n-1)} \cdot \mathbf{\Phi^t(n-1)} + \mathbf{Q(n-1)}$$

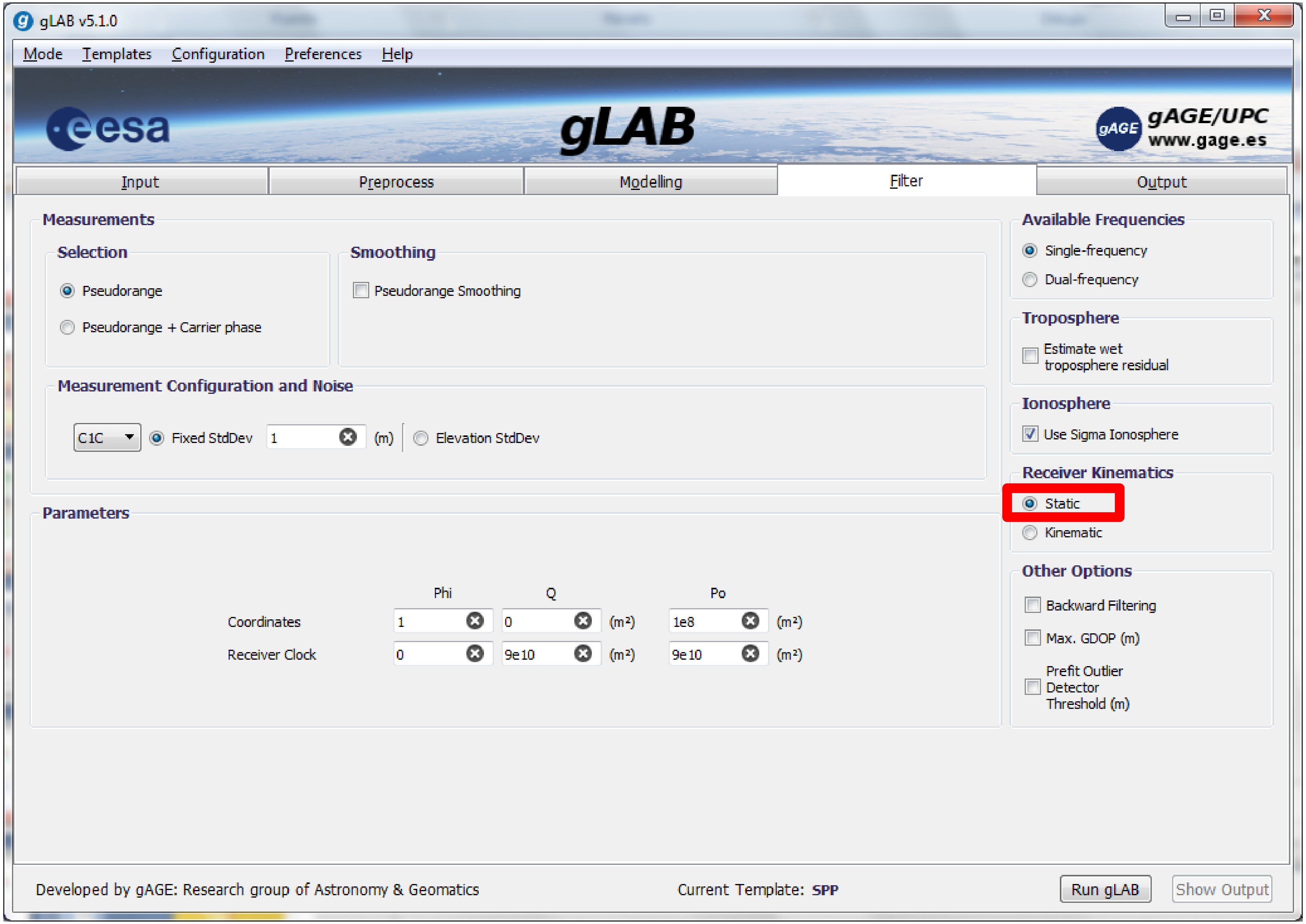
White Noise process N(0, σ)



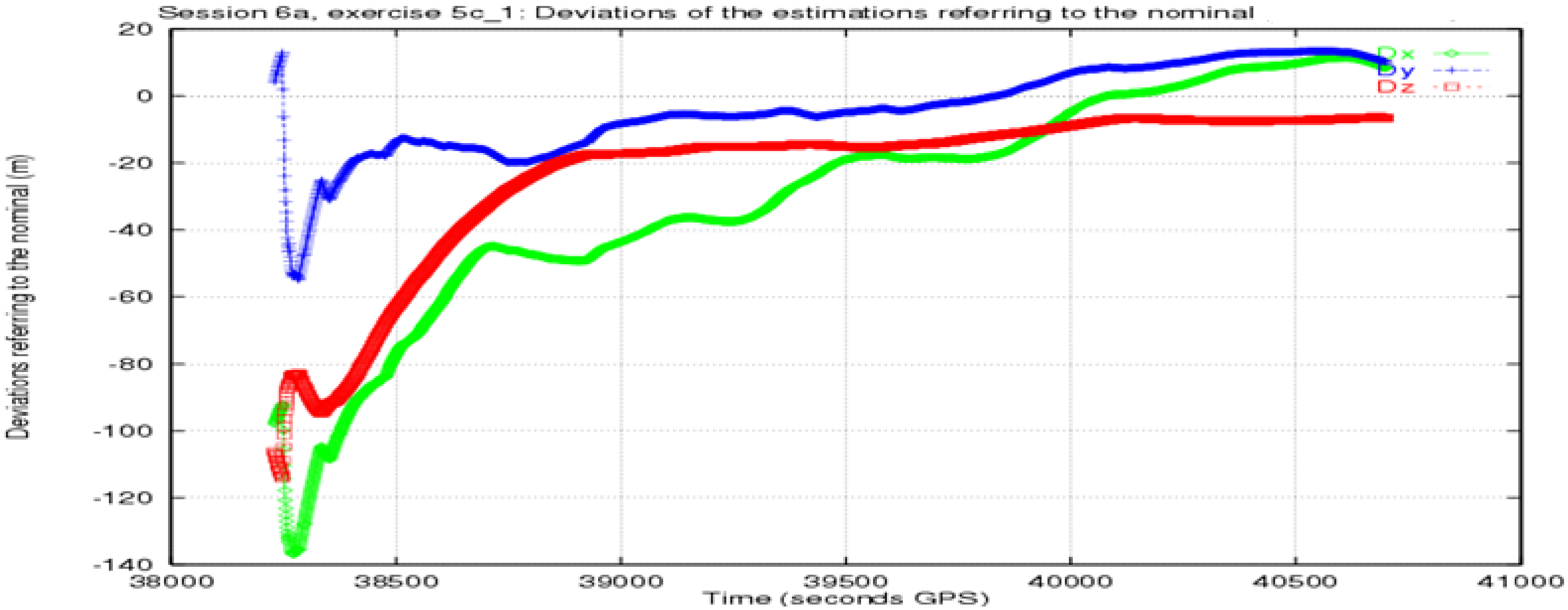
We only can assume that, the next **x(n)** can be **x(n)=0 with a confidence σ.**

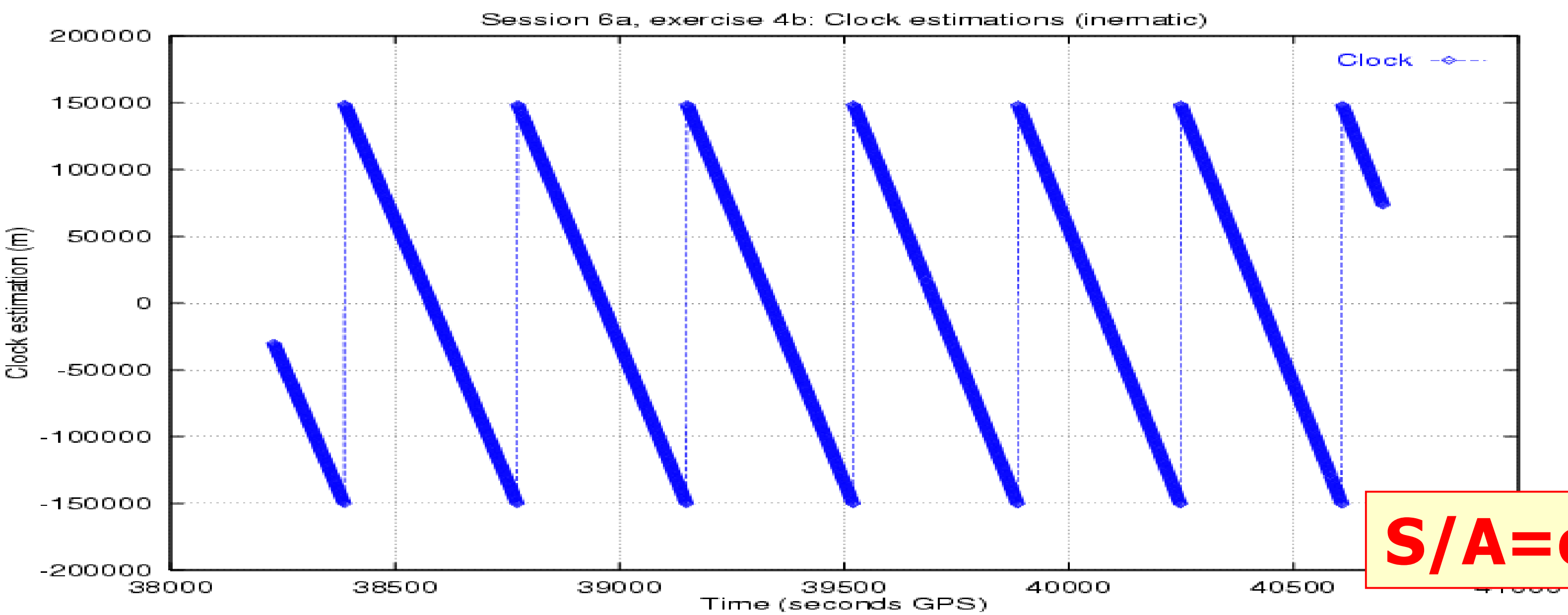
$$\hat{\mathbf{x}}^{-}(n) = \mathbf{\Phi(n-1)} \cdot \hat{\mathbf{x}}(n-1)$$

$$\mathbf{P}_{\hat{\mathbf{x}}^{-}(n)} = \mathbf{\Phi(n-1)} \cdot \mathbf{P}_{\hat{\mathbf{x}}(n-1)} \cdot \mathbf{\Phi^{t}(n-1)} + \mathbf{Q(n-1)}$$


gLAB v5.1.0
Mode
Templates
Configuration
Preferences
Help
esa
gLAB
gAGE/UPC
www.gage.es
Input
Preprocess
Modelling
Filter
Output
Measurements
Selection
Pseudorange
Pseudorange + Carrier phase
Smoothing
Pseudorange Smoothing
Measurement Configuration and Noise
C1C
Fixed StdDev
1
(m)
Elevation StdDev
Parameters
Phi
Q
Po
Coordinates
1
0
(m²)
1e8
(m²)
Receiver Clock
0
9e10
(m²)
9e10
(m²)
Available Frequencies
Single-frequency
Dual-frequency
Troposphere
Estimate wet troposphere residual
Ionosphere
Use Sigma Ionosphere
Receiver Kinematics
Static
Kinematic
Other Options
Backward Filtering
Max. GDOP (m)
Prefit Outlier Detector Threshold (m)
Developed by gAGE: Research group of Astronomy & Geomatics
Current Template: SPP
Run gLAB
Show Output

Static positioning: constant coordinates and white noise clock





S/A=on

gAGE/UPC research group of Astronomy and Geomatics
BarcelonaTECH, Spain

gAGE

gAGE/UPC research group of Astronomy and Geomatics
BarcelonaTECH, Spain

b) Kinematic positioning

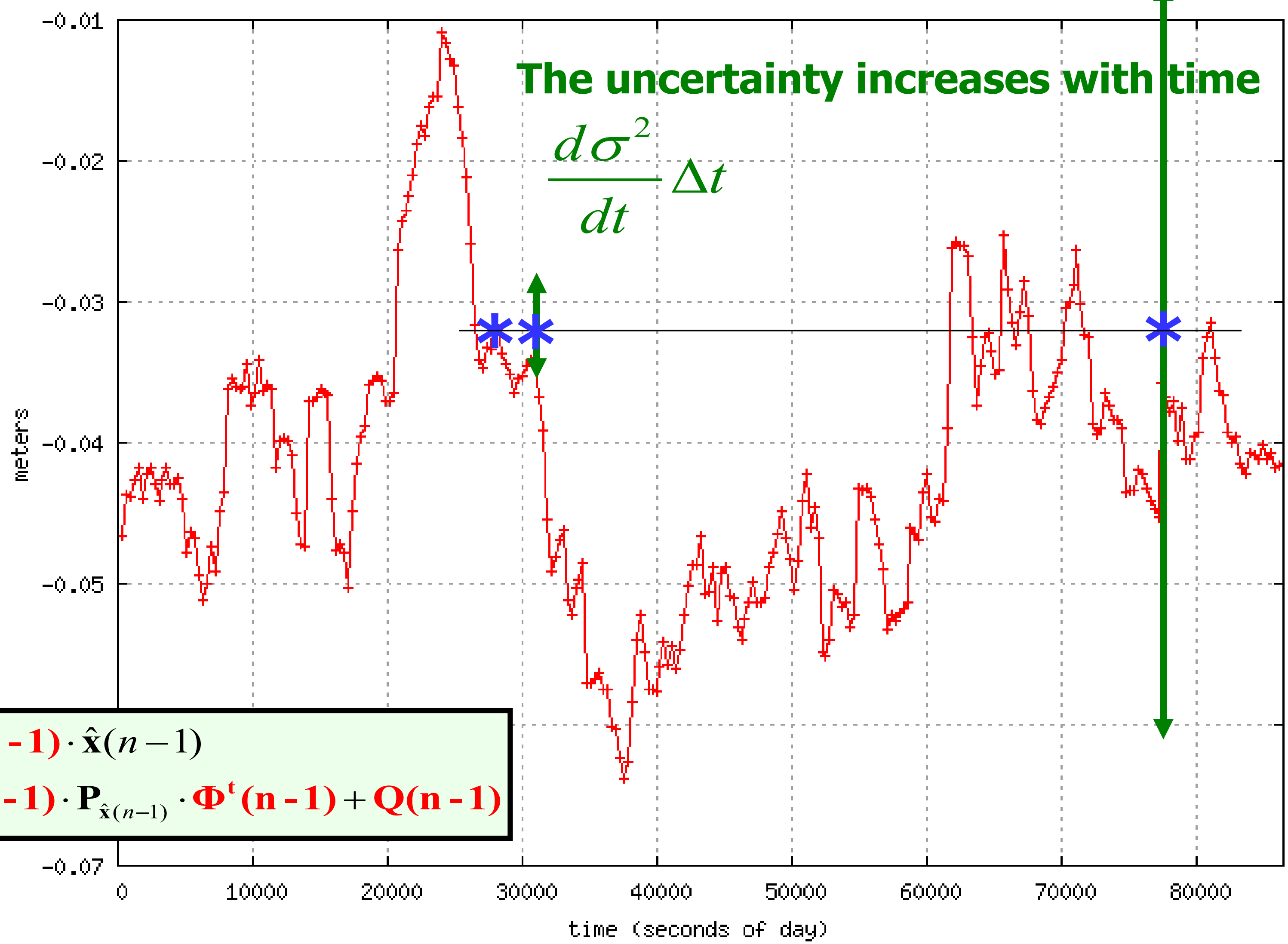
1) In case of a fast moving vehicle, coordinates will be modeled as **white noise** with zero mean, and the same rationale applies for clock offset:

$$\mathbf{\Phi}(n) = \begin{pmatrix} 0 & & & \\ & 0 & & \\ & & 0 & \\ & & & 0 \end{pmatrix} \qquad \mathbf{Q}(n) = \begin{pmatrix} \sigma_{dx}^2 & & & \\ & \sigma_{dy}^2 & & \\ & & \sigma_{dz}^2 & \\ & & & \sigma_{DT}^2 \end{pmatrix}$$

2) In case of a slow moving vehicle, coordinates may be modeled as **random walk**, process' spectral density $\dot{q} = \frac{d\sigma^2}{dt}$, and the clock as a **white noise**:

$$\mathbf{\Phi}(n) = \begin{pmatrix} 1 & & & \\ & 1 & & \\ & & 1 & \\ & & & 0 \end{pmatrix} \qquad \mathbf{Q}(n) = \begin{pmatrix} \dot{q}_{dx}\Delta t & & & \\ & \dot{q}_{dy}\Delta t & & \\ & & \dot{q}_{dz}\Delta t & \\ & & & \sigma_{DT}^2 \end{pmatrix}$$

Random Walk process: it varies slowly



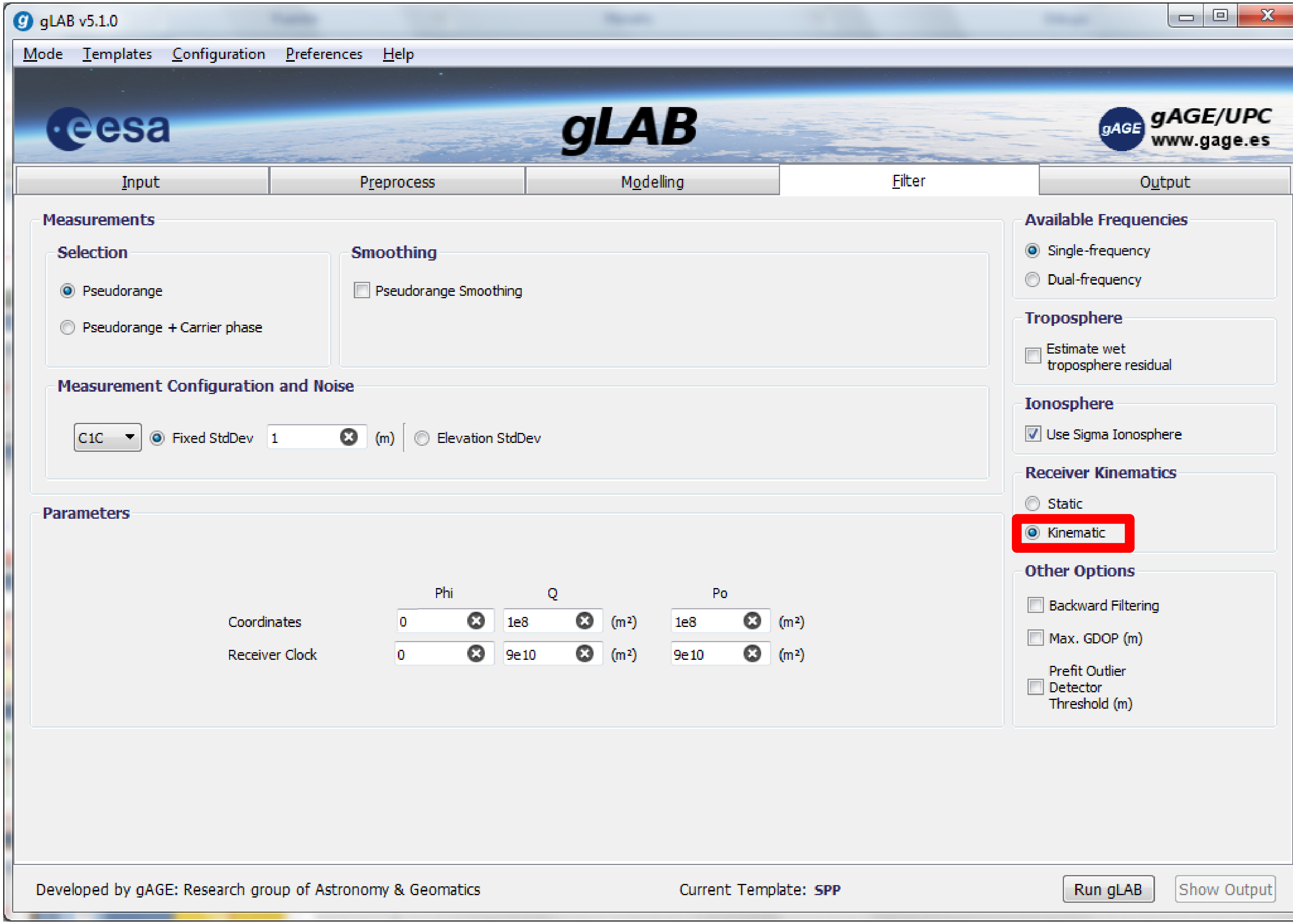
$$\hat{\mathbf{x}}^-(n) = \mathbf{\Phi(n-1)} \cdot \hat{\mathbf{x}}(n-1)$$

$$\mathbf{P}_{\hat{\mathbf{x}}^-(n)} = \mathbf{\Phi(n-1)} \cdot \mathbf{P}_{\hat{\mathbf{x}}(n-1)} \cdot \mathbf{\Phi^t(n-1)} + \mathbf{Q(n-1)}$$

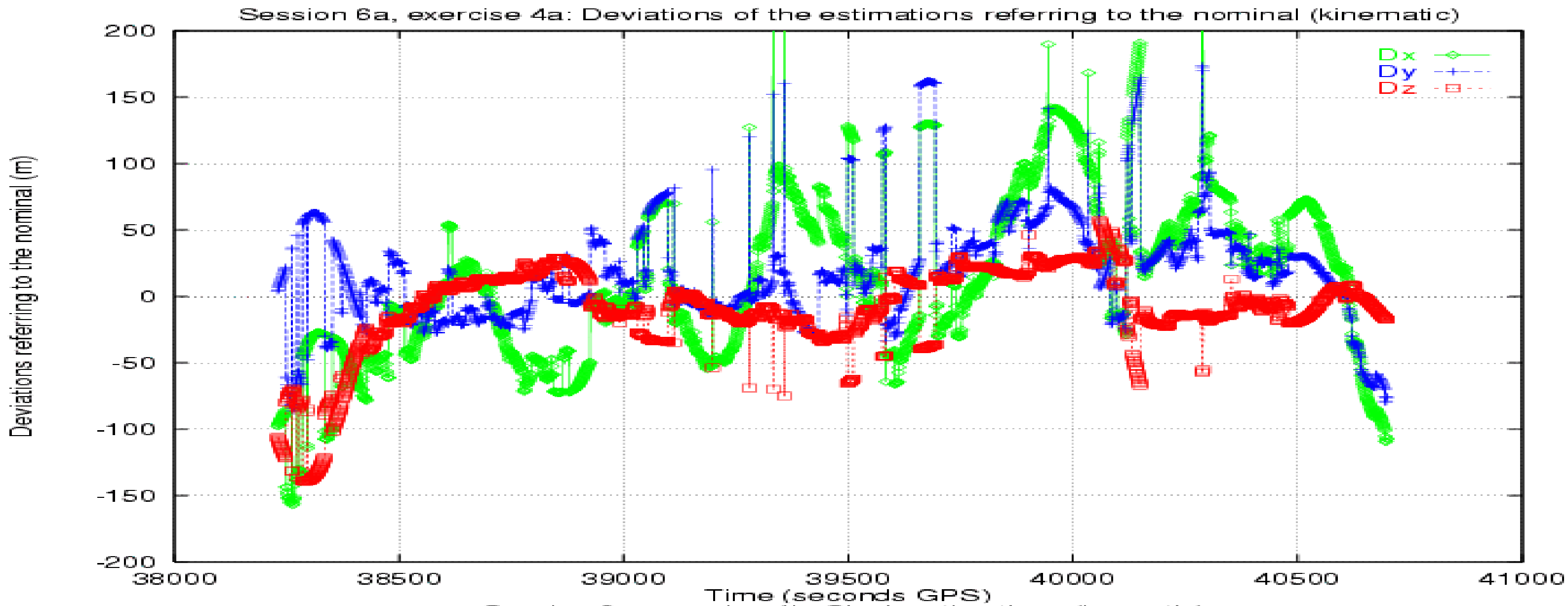
Φ=1 ←→ x(n)=x(n-1) (the same value is assumed)

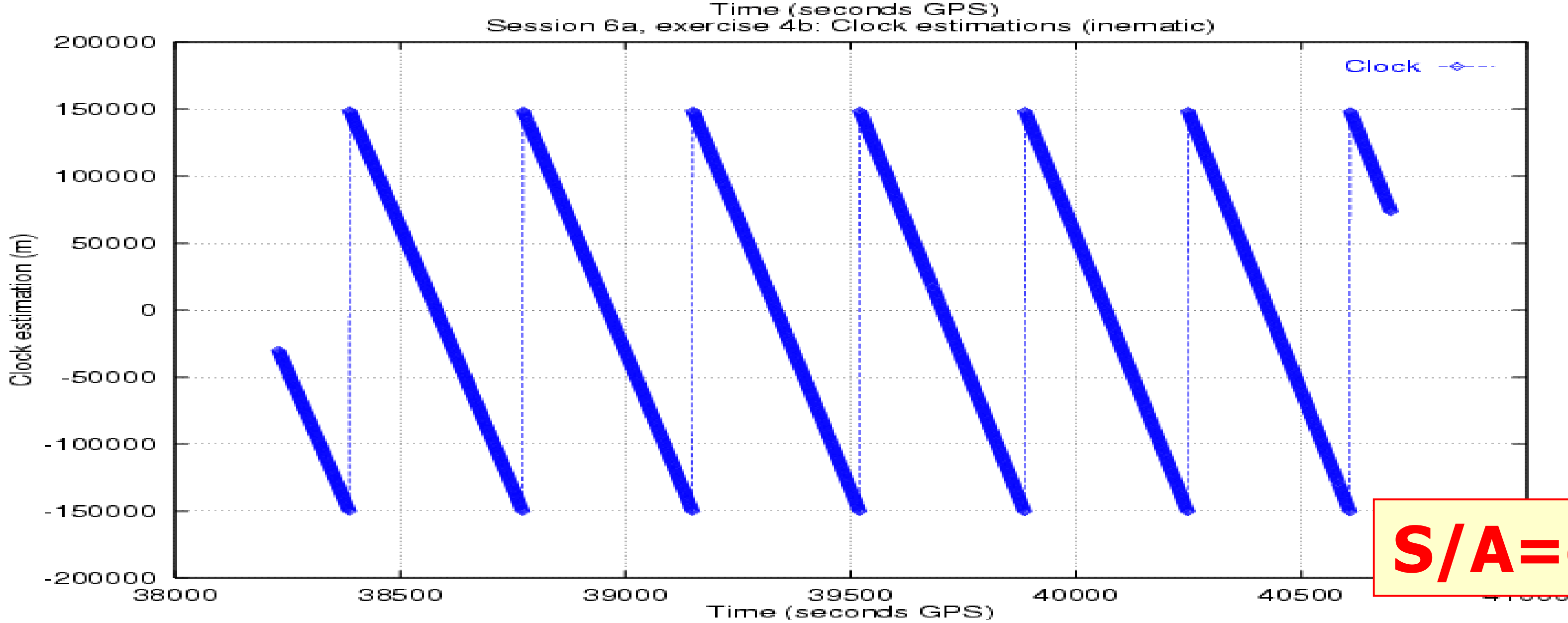
Q=(dσ²/dt)*Δt (but, with prediction error noise increasing with time)

gAGE/UPC research group of Astronomy and Geomatics
BarcelonaTECH, Spain



Pure Kinematic positioning: white noise coordinates and clock





S/A=on

8. Solving with the Kalman filter

The measurement file UPC11490.050 has been collected by a receiver with fixed coordinates. Using navigation file UPC11490.05N, compute the SPP solution in static mode[74] and check *by hand* the computation of the solution for the first three epochs (i.e. $t = 300$, $t = 600$ and $t = 900$ seconds).

Complete the following steps:

(a) Set the default configuration of gLAB for the SPP mode. Then, in section [Filter], select [⊙ Static] in the Receiver Kinematics option. To process the data click Run gLAB.

Solving with the kalman filter (by hand):
See exercise 8, Session 5.2 in [RD-2]

(c) Using the previous equations and the configuration parametres applied by gLAB compute by hand the solution for the first three epochs[75] (i.e. $t = 300$, $t = 600$ and $t = 900$ s).

Note: Use the prefit residual vector $\mathbf{y}(k)$ *and design matrix* $\mathbf{G}(k)$ *computed by* gLAB.

Hint:

i. Filter configuration (according to gLAB):

- Initialisation:

  $\texttt{x0} \equiv \hat{x}(0) = (0, 0, 0, 0)$,

  $\texttt{P0} \equiv \mathbf{P}(0) = \sigma_0^2\,\mathbf{I}$, with $\sigma_0 = 3 \cdot 10^5$m.

- Process noise $\mathbf{Q}$ and transition matrices $\mathbf{\Phi}$:

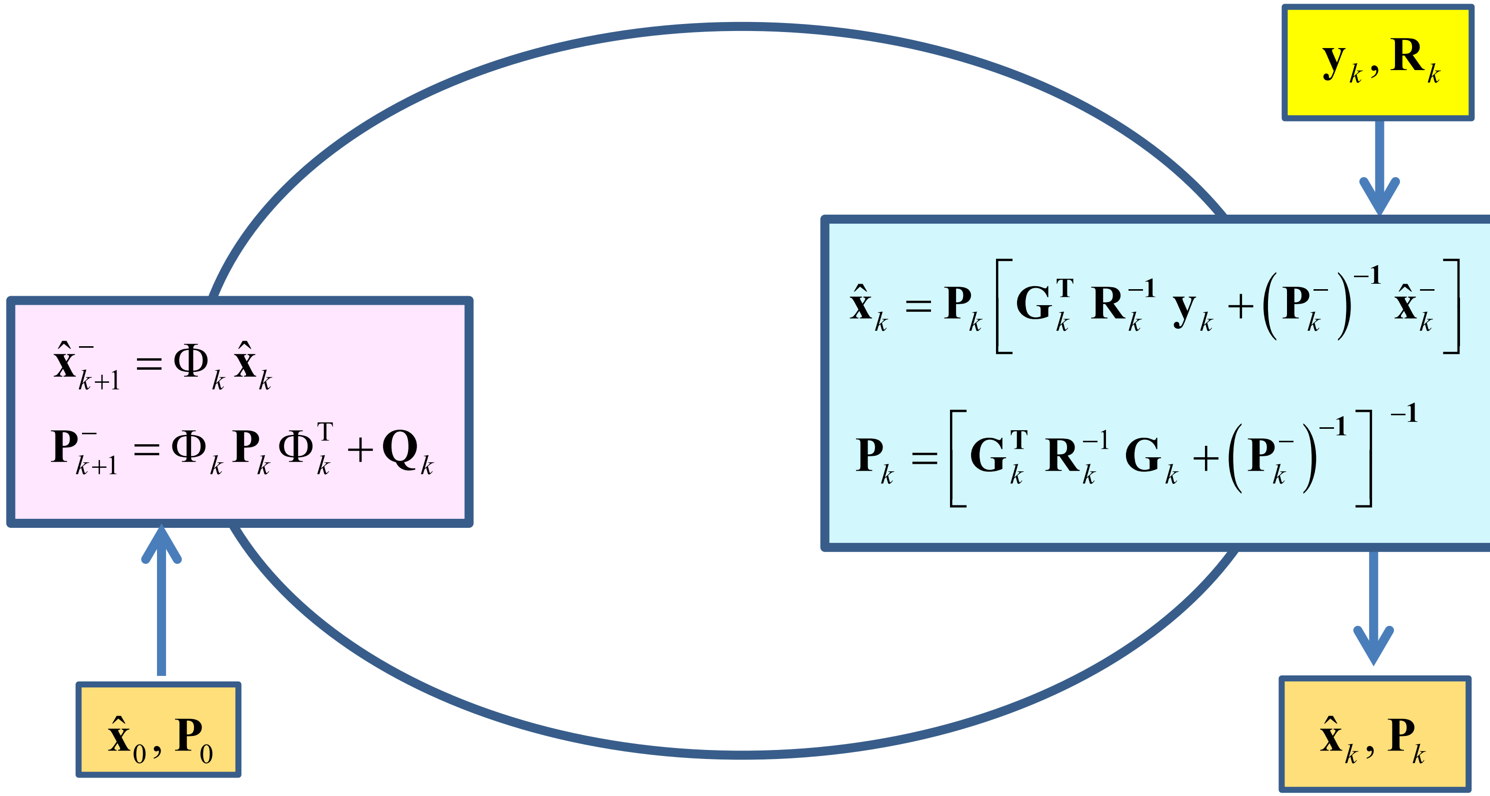
$$\texttt{Q} \equiv Q(k) = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \sigma_{dt}^2 \end{bmatrix}, \quad \Phi \equiv \mathbf{\Phi}(k) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

  with $\sigma_{dt} = 3 \cdot 10^5$m.

- Measurement covariance matrix:

  $\texttt{Rk} \equiv \mathbf{R}(k) = \sigma_y^2\,\mathbf{I}$, with $\sigma_y = 1$ m.

(b) Write the Kalman filter equations according to Fig. 6.2, in section 6.1.2 of Volume I.



ii. Kalman filter iterations:

k=1:

Predict:

$$\mathtt{x1}^- = \Phi \cdot \hat{\mathtt{x}}0$$

$$\mathtt{P1}^- = \Phi \cdot \mathtt{P0} \cdot \Phi^T + \mathtt{Q}$$

Update:

$$\mathtt{P1} = [\mathtt{G1}^T \cdot \mathtt{R1}^{-1} \cdot \mathtt{G1} + (\mathtt{P1}^-)^{-1}]^{-1}$$

$$\hat{\mathtt{x}}1 = \mathtt{P1} \cdot [\mathtt{G1}^T \cdot \mathtt{R1}^{-1} \cdot \mathtt{y1} + (\mathtt{P1}^-)^{-1} \cdot \mathtt{x1}^-]$$

k=2:

Predict:

$$\mathtt{x2}^- = \Phi \cdot \hat{\mathtt{x}}1$$

$$\mathtt{P2}^- = \Phi \cdot \mathtt{P1} \cdot \Phi^T + \mathtt{Q}$$

Update:

$$\mathtt{P2} = [\mathtt{G2}^T \cdot \mathtt{R2}^{-1} \cdot \mathtt{G2} + (\mathtt{P2}^-)^{-1}]^{-1}$$

$$\hat{\mathtt{x}}2 = \mathtt{P2} \cdot [\mathtt{G2}^T \cdot \mathtt{R2}^{-1} \cdot \mathtt{y2} + (\mathtt{P2}^-)^{-1} \cdot \mathtt{x2}^-]$$

k=3:

...

iii. Data vectors and matrices: Vectors $\mathtt{yk} \equiv \mathbf{y}(k)$ and design matrices $\mathtt{Gk} \equiv \mathbf{G}(k)$ are generated from the gLAB.out file.

Execute for instance:[76]

```
grep "PREFIT" gLAB.out | grep -v INFO |
            gawk '{if ($6!=21 )print $0}'|
  gawk '{if ($4==300) print $8,$11,$12,$13,$14}'
                                          > M300.dat

grep "PREFIT" gLAB.out | grep -v INFO |
  gawk '{if ($4==600) print $8,$11,$12,$13,$14}'
                                          > M600.dat

grep "PREFIT" gLAB.out | grep -v INFO |
  gawk '{if ($4==900) print $8,$11,$12,$13,$14}'
                                          > M900.dat
```

Then using Octave or MATLAB:

```
y1=M300(:,1)
G1=M300(:,2:5)

y2=M600(:,1)
G2=M600(:,2:5)

y3=M900(:,1)
G3=M900(:,2:5)
```

iv. Results computed by gLAB:

A. (X,Y,Z) coordinates:

```
grep OUTPUT gLAB.out | grep -v INFO |
         gawk '{if ($4==300) print $9,$10,$11}'

grep OUTPUT gLAB.out | grep -v INFO |
         gawk '{if ($4==600) print $9,$10,$11}'

grep OUTPUT gLAB.out | grep -v INFO |
         gawk '{if ($4==900) print $9,$10,$11}'
```

B. Receiver clock

```
grep FILTER gLAB.out | grep -v INFO |
                    gawk '{if ($4==300) print $8}'

grep FILTER gLAB.out | grep -v INFO |
                    gawk '{if ($4==600) print $8}'

grep FILTER gLAB.out | grep -v INFO |
                    gawk '{if ($4==900) print $8}'
```

Contents

5. Predicted Accuracy: Dilution Of Precision (DOP)

The geometry matrix **G** does not depend on the measurements, then it can be computed even from the almanac (because accurate satellite positions are not needed).

$$\begin{bmatrix} Prefit^1 \\ Prefit^2 \\ \cdots\cdots \\ Prefit^n \end{bmatrix} = \begin{bmatrix} \frac{x_{0,rec}-x^{sat1}}{\rho_{0,rec}^{sat1}} & \frac{y_{0,rec}-y^{sat1}}{\rho_{0,rec}^{sat1}} & \frac{z_{0,rec}-z^{sat1}}{\rho_{0,rec}^{sat1}} & 1 \\ \frac{x_{0,rec}-x^{sat2}}{\rho_{0,rec}^{sat2}} & \frac{y_{0,rec}-y^{sat2}}{\rho_{0,rec}^{sat2}} & \frac{z_{0,rec}-z^{sat2}}{\rho_{0,rec}^{sat2}} & 1 \\ & \cdots\cdots\cdots & & \\ \frac{x_{0,rec}-x^{sat\,n}}{\rho_{0,rec}^{sat\,n}} & \frac{y_{0,rec}-y^{sat\,n}}{\rho_{0,rec}^{sat\,n}} & \frac{z_{0,rec}-z^{sat\,n}}{\rho_{0,rec}^{sat\,n}} & 1 \end{bmatrix} \begin{bmatrix} \Delta x_{rec} \\ \Delta y_{rec} \\ \Delta z_{rec} \\ c\,dt_{rec} \end{bmatrix}$$

G

In this sense the following Dilution Of Precision (DOP) parameters are defined:

$$\mathbf{Q} \equiv (\mathbf{G}^T\,\mathbf{G})^{-1} = \begin{bmatrix} q_{xx} & q_{xy} & q_{xz} & q_{xt} \\ q_{xy} & q_{yy} & q_{yz} & q_{yt} \\ q_{xz} & q_{yz} & q_{zz} & q_{zt} \\ q_{xt} & q_{yt} & q_{zt} & q_{tt} \end{bmatrix}$$

- *Position Dilution Of Precision:*

$$\mathrm{PDOP} = \sqrt{q_{xx} + q_{yy} + q_{zz}}$$

- *Geometric Dilution Of Precision:*

$$\mathrm{GDOP} = \sqrt{q_{xx} + q_{yy} + q_{zz} + q_{tt}}$$

- *Time Dilution Of Precision:*

$$\mathrm{TDOP} = \sqrt{q_{tt}}$$

Predicted Accuracy: Dilution Of Precision (DOP)

The same computation can be done in (e,n,u) coordinates:

$$\begin{bmatrix} Prefit^1 \\ Prefit^2 \\ \cdots\cdots \\ Prefit^n \end{bmatrix} = \underbrace{\begin{bmatrix} -\cos el^1 \sin az^1 & -\cos el^1 \cos az^1 & -\sin el^1 & 1 \\ -\cos el^2 \sin az^2 & -\cos el^2 \cos az^2 & -\sin el^2 & 1 \\ & \cdots\cdots\cdots & & \\ -\cos el^n \sin az^n & -\cos el^n \cos az^n & -\sin el^n & 1 \end{bmatrix}}_{\mathbf{G}} \begin{bmatrix} \Delta e_{rec} \\ \Delta n_{rec} \\ \Delta u_{rec} \\ c\,dt_{rec} \end{bmatrix}$$

$$\mathbf{Q} \equiv (\mathbf{G}^T\,\mathbf{G})^{-1}$$

- *Horizontal Dilution Of Precision:*

$$\text{HDOP} = \sqrt{q_{ee} + q_{nn}}$$

- *Vertical Dilution Of Precision:*

$$\text{VDOP} = \sqrt{q_{uu}}$$

Hence, estimations of the expected accuracy are given by

GDOP σ	geometric precision in position and time
PDOP σ	precision in position
TDOP σ	precision in time
HDOP σ	precision in horizontal positioning
VDOP σ	precision in vertical positioning

gAGE

gAGE/UPC research group of Astronomy and Geomatics
BarcelonaTECH, Spain

For all sat. in view

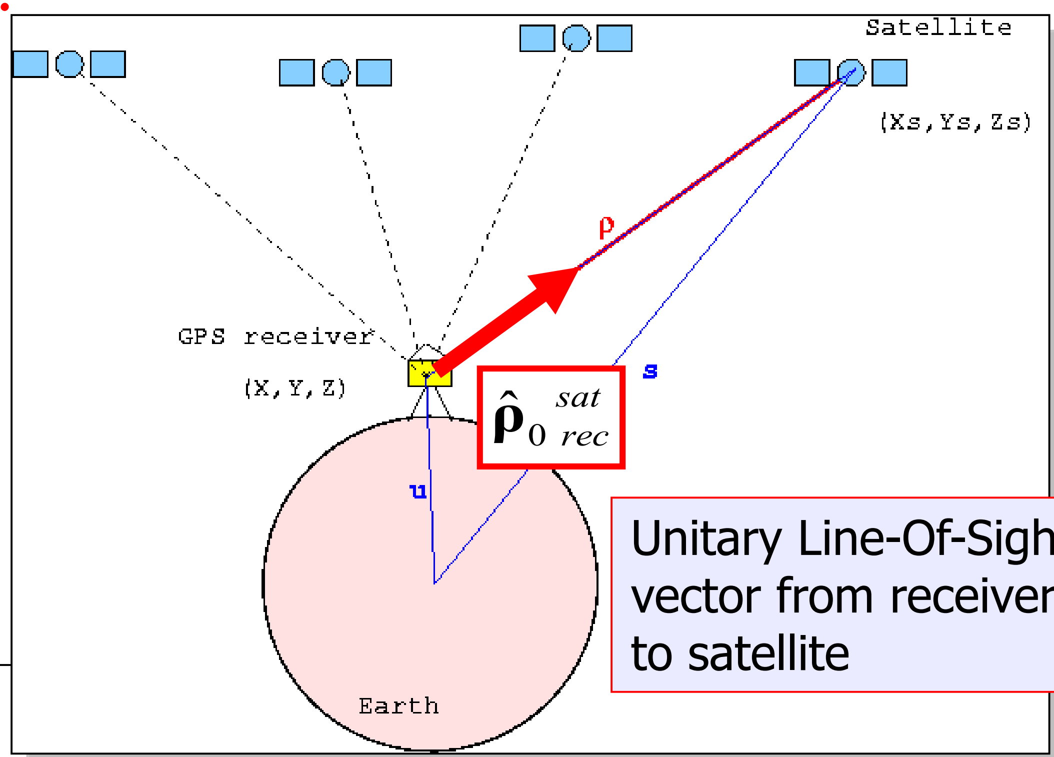
$$
\begin{bmatrix} Prefit^1 \\ Prefit^2 \\ \cdots\cdots \\ Prefit^n \end{bmatrix}
=
\begin{bmatrix}
\frac{x_{0,rec}-x^{sat1}}{\rho_{0,rec}^{sat1}} & \frac{y_{0,rec}-y^{sat1}}{\rho_{0,rec}^{sat1}} & \frac{z_{0,rec}-z^{sat1}}{\rho_{0,rec}^{sat1}} & 1 \\
\frac{x_{0,rec}-x^{sat2}}{\rho_{0,rec}^{sat2}} & \frac{y_{0,rec}-y^{sat2}}{\rho_{0,rec}^{sat2}} & \frac{z_{0,rec}-z^{sat2}}{\rho_{0,rec}^{sat2}} & 1 \\
 & \cdots\cdots\cdots & & \\
\frac{x_{0,rec}-x^{sat\,n}}{\rho_{0,rec}^{sat\,n}} & \frac{y_{0,rec}-y^{sat\,n}}{\rho_{0,rec}^{sat\,n}} & \frac{z_{0,rec}-z^{sat\,n}}{\rho_{0,rec}^{sat\,n}} & 1
\end{bmatrix}
\begin{bmatrix} \Delta x_{rec} \\ \Delta y_{rec} \\ \Delta z_{rec} \\ c\,dt_{rec} \end{bmatrix}
$$

Geometry of rays

Observations
(measured-computed)

$$-\frac{\boldsymbol{\rho}_{0\ rec}^{T\ sat\,n}}{\rho_{0\ rec}^{sat\,n}}$$

$$\hat{\boldsymbol{\rho}}_{0\ rec}^{T\ sat\,n} \equiv \frac{\boldsymbol{\rho}_{0\ rec}^{T\ sat\,n}}{\rho_{0\ rec}^{sat1}}$$



$$\hat{\boldsymbol{\rho}}_{0\ rec}^{\ sat}$$

Unitary Line-Of-Sight vector from receiver to satellite

References

[RD-1] J. Sanz Subirana, J.M. Juan Zornoza, M. Hernández-Pajares, GNSS Data processing. Volume 1: Fundamentals and Algorithms. ESA TM-23/1. ESA Communications, 2013.

[RD-2] J. Sanz Subirana, J.M. Juan Zornoza, M. Hernández-Pajares, GNSS Data processing. Volume 2: Laboratory Exercises. ESA TM-23/2. ESA Communications, 2013.

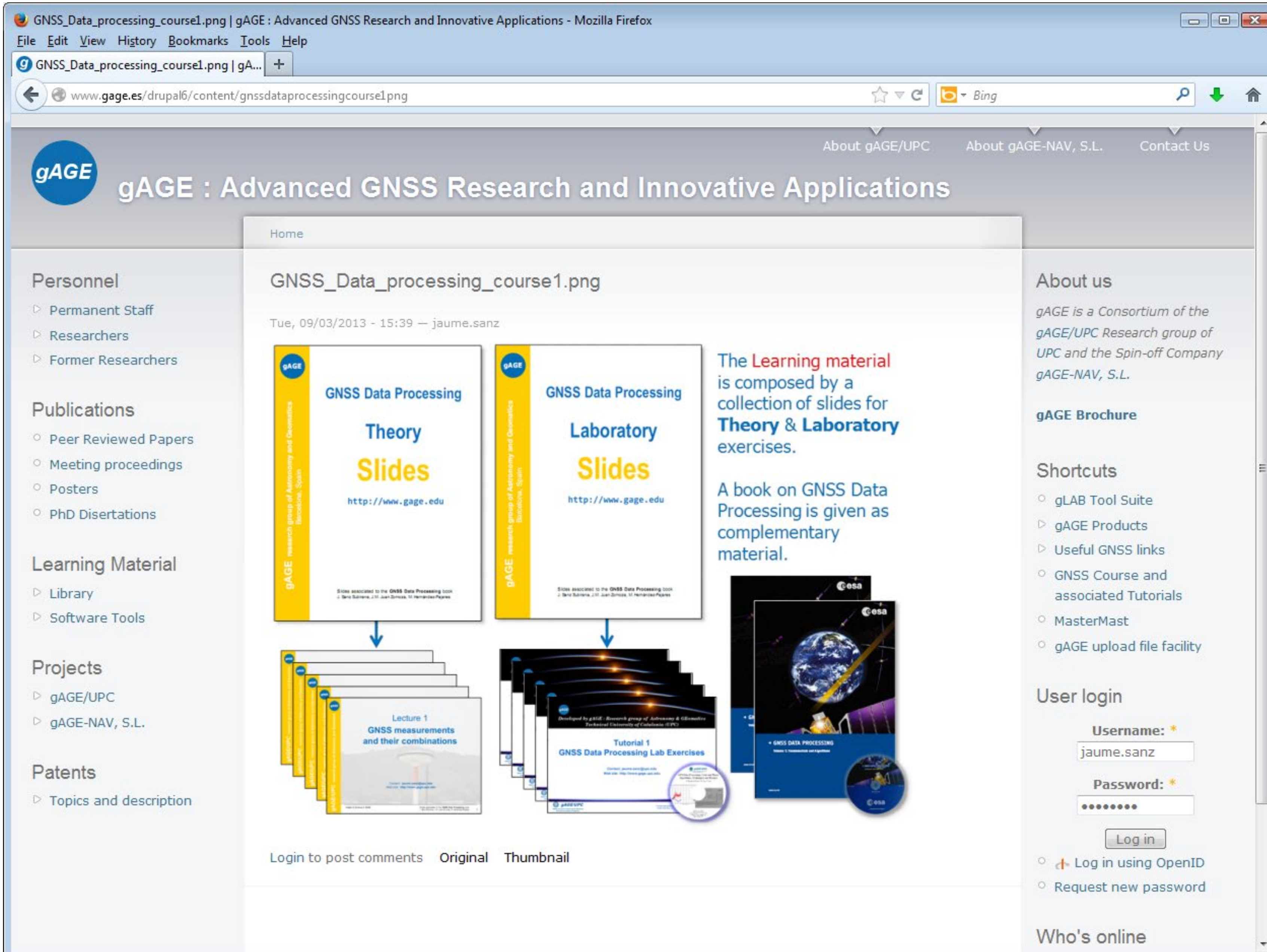
[RD-3] Pratap Misra, Per Enge. Global Positioning System. Signals, Measurements, and Performance. Ganga –Jamuna Press, 2004.

[RD-4] B. Hofmann-Wellenhof et al. GPS, Theory and Practice. Springer-Verlag. Wien, New York, 1994.

Thank you

gAGE

gAGE/UPC research group of Astronomy and Geomatics BarcelonaTECH, Spain



gAGE

gAGE/UPC research group of Astronomy and Geomatics BarcelonaTECH, Spain

GNSS Data Processing, Vol. 1: Fundamentals and Algorithms.
GNSS Data Processing, Vol. 2: Laboratory exercises.