GNSS PVT Lab

Analysis of WLS and KF-based Navigation from GPS Observables (Static and Dynamic Rx)

Gaël Pagès
gael.pages@isae-supaero.fr

2021

1 Introduction

The main goal of this Lab is to **understand how to estimate a position, velocity and timing (PVT)** solution from signals transmitted by the GPS constellation, using observables from a single-frequency GPS receiver (**real signal**). Collaterally, this implies to analyse and understand:

- the impact of the receiver dynamics,
- observable corrections,
- and satellite constellation geometry.

Figure 1 illustrates the overview of a GNSS receiver. In this study we will be interested in the **"navigator"** which has the role of estimating the absolute position, velocity and clock bias of the receiver from the measurements provided (GNSS observables) by the baseband signal processing stage.

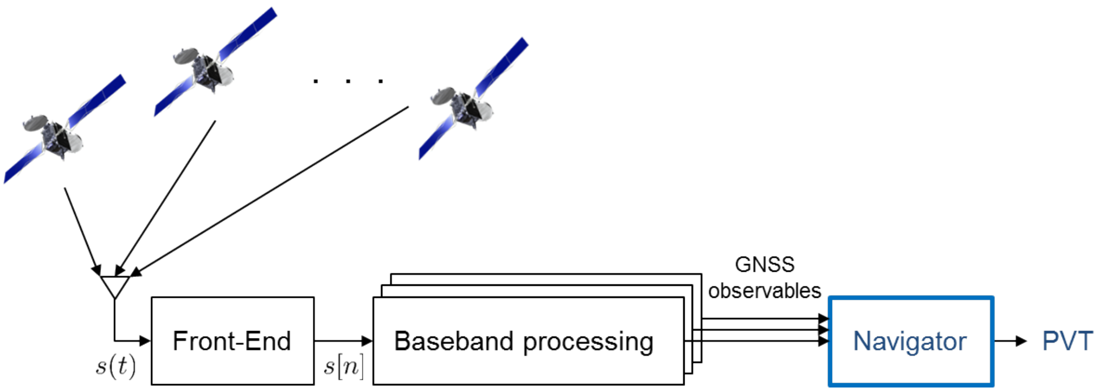

Figure 1: General GNSS receiver architecture.

In this study, for N_s satellites in view, the **GNSS observables available** are,

- ρ_i: **pseudoranges** (m), for $i = 1 \dots N_s$
- f_{d_i}: **Doppler frequencies** (Hz), for $i = 1 \dots N_s$

In addition, we also have access to,

- $C/N0_i$: carrier-to-noise density ratio (dB-Hz)
- the **navigation message** for each visible satellite

The $C/N0$ gives an indication of the quality of the received signal in a GNSS receiver. For a GPS L1 C/A receiver the typical values of $C/N0$ vary between 35 dB-Hz and 45 dB-Hz. For values below 30 dB-Hz we may consider that the signal is severely degraded.

The Doppler frequency gives information about the radial velocity of the satellite, so-called pseudovelocity or **pseudorange rate**, $\dot{\rho}_i$,

$$\dot{\rho}_i = -\frac{c}{f_{L1}} f_{d_i}, \tag{1}$$

where $c \approx 2.99792458e8$ m/s is the speed of light and $f_{L1} = 1575,42$ MHz is the carrier frequency of the signal transmitted on the L1 band.

The pseudorange and pseudorange rate measurements can be modeled by the following equations,

$$\rho_i = ||\mathbf{p}_u - \mathbf{p}_i|| + b_u - c\delta t_{s_i} - c\delta t_{r_i} + c\delta t_{I_i} + c\delta t_{T_i} + ct_{GD_i} + \varepsilon_{\rho_i} \tag{2}$$

$$\dot{\rho}_i = \mathbf{los}_i{}^T(\mathbf{v}_u - \mathbf{v}_i) + d_u - c\dot{\delta t}_{s_i} + \varepsilon_{\dot{\rho}_i} \tag{3}$$

with the following notation:

$\mathbf{p}_u$	Receiver position in the ECEF coordinate frame
$\mathbf{p}_i$	i-th satellite position in the ECEF coordinate frame
b_u	Receiver clock bias (m)
$c\delta t_{s_i}$	i-th satellite clock bias (m)
$c\delta t_{r_i}$	i-th satellite relativistic clock bias (m)
$c\delta t_{I_i}$	Ionospheric delay (m)
$c\delta t_{T_i}$	Tropospheric delay (m)
t_{GD_i}	i-th satellite instrumental group delay (for single frequency Rx)
ε_{ρ_i}	Noise and unmodeled pseudorange errors
$\mathbf{v}_u$	Receiver velocity in the ECEF coordinate frame
$\mathbf{v}_i$	i-th satellite velocity in the ECEF coordinate frame
d_u	Receiver clock drift (m/s)
$c\dot{\delta t}_{s_i}$	i-th satellite clock drift (m/s)
$\varepsilon_{\dot{\rho}_i}$	Noise and unmodeled pseudorange rate errors

Table 1: Notations.

Notice that the receiver clock bias and clock drift are already given in units of meters and m/s. Also when dealing with real signals, we must correctly take into account all the possible errors: ionospheric, tropospheric, relativistic and group delays.

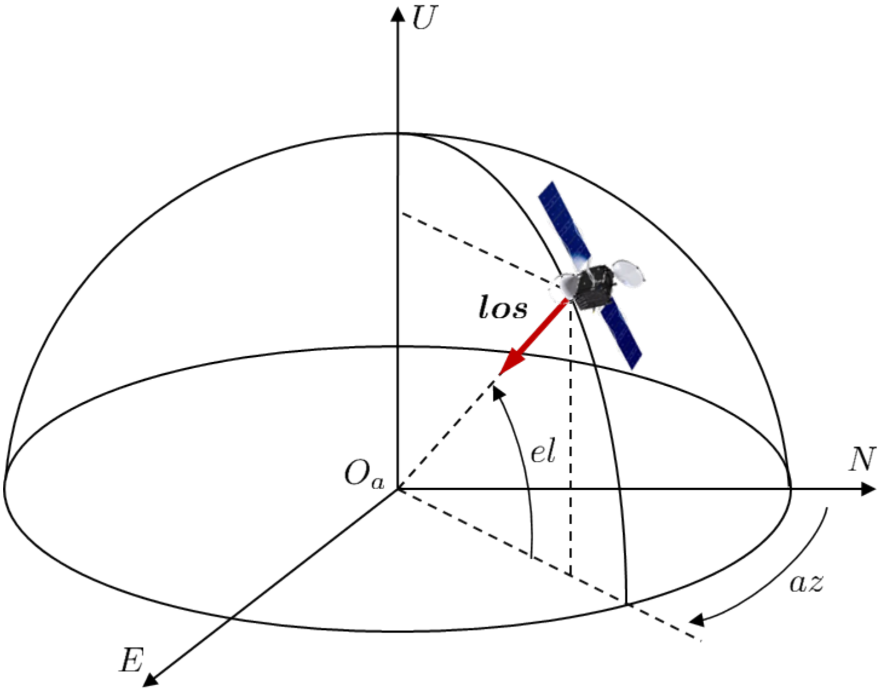

Figure 2: Elevation (el), azimuth (az) and **los** vector for one satellite in the local ENU (East, North, Up) coordinate frame.

The **los** (*Line-Of-Sight*) vector defines the satellite to receiver unit vector (steering vector). It is shown in Figure 2 and is mathematically defined as:

$$\mathbf{los}_i = \frac{1}{||\mathbf{p}_u - \mathbf{p}_i||} \begin{bmatrix} x_u - x_i & y_u - y_i & z_u - z_i \end{bmatrix}^T \tag{4}$$

The satellite clock bias can be computed from the data in the navigation message. This provides clock information in the form of coefficients of a polynomial, in a given reference epoch, t_{oc} (*time of clock*). The clock bias (s) is defined as follows

$$\delta t_{s_i} = a_{f0} + a_{f1}(t - t_{oc}^i) + a_{f2}(t - t_{oc}^i)^2 \tag{5}$$

where t represents the GPS time, *Time of Week* (TOW), of the satellite.

2 Static Receiver Positioning

In this first part of the Lab, we will be interested in the analysis and performance of an estimator allowing to obtain the position of a static receiver (fixed point).

- Open MATLAB
- Set the path to **be_gnss_matlab_etu**.
- Open the script **main_gnss.m** and verify that **Flag_spp** is equal to 1 (line 22).

In this folder you will find the following:

- **data**: GNSS measurements recorded in an open sky environment
- **lib_coordinates_conversion**: coordinate frame transformation libs
- **lib_gps**: GPS corrections and satellite position computation libs
- **navigator**: navigation algorithms
- **utilities**: utility functions
- **main_gnss.m** : the main script for the Lab
- **user_config.m** : user config file

2.1 Observation model

The state vector $\mathbf{x}$ with the set of parameters to be estimated is given by

$$\mathbf{x} = \begin{bmatrix} \mathbf{p}_u^T & \mathbf{v}_u^T & b_u & d_u \end{bmatrix}^T \tag{6}$$

In the Matlab scripts, the following indexes are defined for the state vector:

- **i_pos = 1:3** : index of the position vector
- **i_vel = 4:6** : index of the velocity vector
- **i_cb = 7** : index of the clock bias
- **i_cd = 8** : index of the clock drift

To estimate the fixed point position we use a nonlinear weighted least squares (**WLS**) solution. This corresponds to the minimisation of the following cost function

$$J = \left(\tilde{\mathbf{z}} - \mathbf{z}(\mathbf{x})\right)^T \mathbf{R}^{-1} \left(\tilde{\mathbf{z}} - \mathbf{z}(\mathbf{x})\right) \tag{7}$$

with $\tilde{\mathbf{z}} = \begin{bmatrix} \tilde{\boldsymbol{\rho}} & \tilde{\dot{\boldsymbol{\rho}}} \end{bmatrix}^T$ the receiver measurements available and $\mathbf{z}(\mathbf{x}) = \begin{bmatrix} \boldsymbol{\rho}(\mathbf{x}) & \dot{\boldsymbol{\rho}}(\mathbf{x}) \end{bmatrix}^T$ the pseudorange and pseudorange rate observation models in (2) et (3), respectively. $\mathbf{R}$ is the measurement noise covariance matrix $\mathbf{R} = \mathrm{diag}(\sigma^2_{\rho_1}, \ldots, \sigma_{\rho^2_{N_s}}, \sigma^2_{\dot{\rho}_1}, \ldots, \sigma^2_{\dot{\rho}_{N_s}})$.

In this study, the pseudorange and pseudorange rate noise variances, σ_ρ^2 and $\sigma_{\dot\rho}^2$, are computed from the (estimated) $C/N0$ as

$$
\begin{aligned}
\sigma_{\rho_i}^2 &= \sigma_\rho^2 \cdot 10^{((30-CN0_i)/10)} && (8)\\
\sigma_{\dot\rho_i}^2 &= \sigma_{\dot\rho}^2 \cdot 10^{((30-CN0_i)/10)} && (9)
\end{aligned}
$$

defined for the i-th satellite at 30 dB-Hz. In the Lab these are assumed to be known values and are defined in the file **user_config.m**.

The solution to the WLS minimization problem is obtained by a first order linearisation of equations (2) and (3). We obtain the following recursive solution,

$$
\hat{\mathbf{x}}_k = \hat{\mathbf{x}}_{k-1} + \left\{ \left(\mathbf{H}_{k-1}^T(\mathbf{x}) \mathbf{R}_k^{-1} \mathbf{H}_{k-1}(\mathbf{x}) \right)^{-1} \mathbf{H}_{k-1}^T(\mathbf{x}) \mathbf{R}_k^{-1} \Delta\mathbf{z}_k \right\}_{\mathbf{x}=\hat{\mathbf{x}}_{k-1}} \tag{10}
$$

where $\hat{\mathbf{x}}_k$ is the estimate of $\mathbf{x}$ at time instant k, $\Delta\mathbf{z}_k = \tilde{\mathbf{z}}_k - \mathbf{z}(\hat{\mathbf{x}}_{k-1})$ is the innovation at k and $\mathbf{H}_{k-1}(\mathbf{x}) = \partial\mathbf{z}(\mathbf{x}_{k-1})/\partial\mathbf{x}$ is the Jacobian matrix, also called observation matrix. Its dimension is $2N_s \times N_x$, N_x being the state vector dimension. For this study, the observation matrix can be expressed (for the i-th satellite) as

$$
\mathbf{H}^i(\mathbf{x}) = \begin{bmatrix} \partial\rho_i/\partial\mathbf{p}_u & \partial\rho_i/\partial\mathbf{v}_u & \partial\rho_i/\partial b_u & \partial\rho_i/\partial d_u \\ \partial\dot\rho_i/\partial\mathbf{p}_u & \partial\dot\rho_i/\partial\mathbf{v}_u & \partial\dot\rho_i/\partial b_u & \partial\dot\rho_i/\partial d_u \end{bmatrix} \tag{11}
$$

with

$$
\begin{aligned}
\partial\rho_i/\partial\mathbf{p}_u &= \begin{bmatrix} \partial\rho_i/\partial x_u & \partial\rho_i/\partial y_u & \partial\rho_i/\partial z_u \end{bmatrix}\\
\partial\rho_i/\partial\mathbf{v}_u &= \begin{bmatrix} \partial\rho_i/\partial v_{u,x} & \partial\rho_i/\partial v_{u,y} & \partial\rho_i/\partial v_{u,z} \end{bmatrix}\\
\partial\dot\rho_i/\partial\mathbf{p}_u &= \begin{bmatrix} \partial\dot\rho_i/\partial x_u & \partial\dot\rho_i/\partial y_u & \partial\dot\rho_i/\partial z_u \end{bmatrix}\\
\partial\dot\rho_i/\partial\mathbf{v}_u &= \begin{bmatrix} \partial\dot\rho_i/\partial v_{u,x} & \partial\dot\rho_i/\partial v_{u,y} & \partial\dot\rho_i/\partial v_{u,z} \end{bmatrix}
\end{aligned}
$$

Exercise 1: The observation matrix

1. From equations (2) and (3) compute the observation matrix $\mathbf{H}^i(\mathbf{x})$ for one observation i. Consider that the **los** vector **variation with respect to position can be assumed negligible**.

2. Give the observation matrix as function of the **los** vector, for one observation i.

3. Taking into account that the complete observation matrix is given by

$$\mathbf{H}(\mathbf{x}) = \begin{bmatrix} \partial\rho_1(\mathbf{x})/\partial\mathbf{x} \\ \vdots \\ \partial\rho_{N_s}(\mathbf{x})/\partial\mathbf{x} \\ \partial\dot{\rho}_1(\mathbf{x})/\partial\mathbf{x} \\ \vdots \\ \partial\dot{\rho}_{N_s}(\mathbf{x})/\partial\mathbf{x} \end{bmatrix}$$

   complete lines 102 to 105 from the script **navigator/gnss__nav__wlmse.m**.

4. Run the script **main__gnss.m** and **comment the result** (Matlab figures 1 and 2).

2.2 Corrections

Exercice 2: GPS corrections

1. Obtain the expression of the satellite clock drift from eq. (5).
2. Open the script **lib_gps>satellite_clock.m**.
3. Complete lines 47/48 to compute the satellite clock terms.
4. Open the script **navigator>PR_estimation.m**.
5. Complete lines 90 to 111 in order to apply all the pseudo-range correction (i.e. satellite clock bias, t_{GD}, relativisitic, iono and tropo delays). Notice that you have the functions **delay_iono.m** and **delay_tropo.m**. The group delay is in the ephemeris structure.
6. In **main_gnss.m**, line 23, keep **Flag_propag** equal to 0, that is, no iono and tropo corrections included.
7. Run the script **main_gnss.m** and **comment the result** (Matlab figures 1 and 2).
8. Set **Flag_propag** equal to 1, that is, we apply the iono and tropo corrections.
9. Run the script **main_gnss.m** again and **comment the result** (Matlab figures 1 and 2).

2.3 Performance analysis

Lines 175 to 177 of the script **main_gnss.m** define the positioning error standard deviation (East, North, Up) at every time instant. These standard deviations are obtained from the estimation error covariance $\mathbf{P}$,

$$\mathbf{P} = \left(\mathbf{H}^T(\mathbf{x})\mathbf{R}^{-1}\mathbf{H}(\mathbf{x})\right)^{-1} \tag{12}$$

Then, $\sigma_{ee} = \sqrt{\boldsymbol{P}(1,1)}$, $\sigma_{nn} = \sqrt{\boldsymbol{P}(2,2)}$ et $\sigma_{uu} = \sqrt{\boldsymbol{P}(3,3)}$ provide us with a measure of the theoretical performance of the estimator.

Exercice 3: Theoretical estimation error

1. Looking at the Matlab figure 2, are the errors correctly limited to 3-σ?

In order to properly characterise the estimator we use the Root Mean Square Error (RMSE) defined as

$$\mathrm{RMSE}(\hat{\mathbf{x}}) = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(\hat{\mathbf{x}}_i - \mathbf{x}_i)^2} \tag{13}$$

Exercice 4: Performance

1. Give the mathematical expression of the RMSE for the East, North and horizontal (EN) error.
2. Complete the script **main_gnss.m**, lines 183 to 185, with these 3 RMSE expressions.
3. Run the main script. Which are the values of the East, North and horizontal RMSE? **Comment the result.**

In order to analyse the impact of the satellite constellation geometry, we will compare two different constellations:

$$\text{SVID} = [4\ 8\ 11\ 20\ 22] \quad \text{et} \quad \text{SVID} = [8\ 10\ 11\ 18\ 20]$$

Exercice 5: On the errors due to the constellation geometry

1. In the main script **main_gnss.m**, line 24, set **Flag_nsv** to 1 (also keep **Flag_propag** equal to 1).
2. Set the SVID vector, line 77, to one of the aforementioned constellations and run the main script. Do the same with the other constellation.
3. Which constellation gives better results and why?

Exercice 6: More on propagation errors

1. Just looking at the skyplot and Matlab figures 3 and 4, can you explain why satellite 14 is more affected by the propagation errors, and why satellite 8 is less affected?
2. In the script **main_gnss.m**, line 24, set **Flag_nsv** back to 0 and run again.
3. Regarding satellite 28, looking at Matlab figures 4 and 5, together with skyplot, how do you explain its behaviour?
4. Set **Flag_nsv** to 0 for the sequel.

Exercice 7: Google Earth plot

1. In the script **main_gnss.m**, set **Flag_kml** to 1 (line 25), and run the code.
2. The user-defined initial position and the corresponding estimated position are displayed in Google Earth. What do you notice?

3 Dynamic Receiver Positioning

In this second part of the Lab, we will implement an Extended Kalman filter (EKF) in order to take into account the vehicle (receiver) dynamics.

3.1 Brief Kalman filter background

We first consider a state-space representation of the system dynamics

$$\begin{cases} \mathbf{x}_k &= \mathbf{f}(\mathbf{x}_{k-1}) + \mathbf{w}_{k-1} \\ \mathbf{z}_k &= \mathbf{h}(\mathbf{x}_k) + \mathbf{v}_k \end{cases} . \tag{14}$$

The Kalman filter (KF) provides a recursive way to compute an estimate of the states of the system, $\mathbf{x}_k$, from a set of noisy observations $\{\mathbf{z}_k\}_{k=1}^{k=N}$, and considering a state-space system model. The KF has a prediction stage and an update or correction stage. The prediction stage uses the estimated state at the previous time instant $(k-1)$ to produce an estimate (prediction) of the current state using the state evolution equation (prediction of the state at k from the estimate at $k-1$). In the update step, the current observations are used to correct the predicted state in order to obtain a better estimate. If the system is nonlinear (as in (14)), in order to use the standard KF equations we have to linearize the state and observation functions f and h. In such case, we obtain the observation matrix $\mathbf{H}$ (dimension $2N_s \times N_x$) and state evolution matrix $\mathbf{F}$ (dimension $N_x \times N_x$) by computing the partial derivative w.r.t. the state (a.k.a Jacobian matrix), and evaluate them at the closest estimate of the true state: i) $\mathbf{F}$ evaluated at the previous estimate $\hat{\mathbf{x}}_{k-1|k-1}$, and ii) $\mathbf{H}$ evaluated at the predicted state $\hat{\mathbf{x}}_{k|k-1}$.

$$\mathbf{F}_{k-1} = \left.\frac{\partial f(\mathbf{x}_{k-1})}{\partial \mathbf{x}_{k-1}}\right|_{\mathbf{x}=\hat{\mathbf{x}}_{k-1|k-1}} \qquad \mathbf{H}_k = \left.\frac{\partial h(\mathbf{x}_k)}{\partial \mathbf{x}_k}\right|_{\mathbf{x}=\hat{\mathbf{x}}_{k|k-1}}$$

The so-called Extended KF (EKF) equations are detailed in table 2.

***Note**: Unlike the linear KF, the EKF does not guarantee the global convergence because of the local linearization. The stability of this filter is therefore more difficult to guarantee and often depends on its correct initialization.*

Prediction			
$\hat{\mathbf{x}}_{k\|k-1}$	=	$f(\hat{\mathbf{x}}_{k-1\|k-1}) + \mathbf{w}_{k-1}$	Predicted state
$\mathbf{P}_{k\|k-1}$	=	$\mathbf{F}_{k-1}\mathbf{P}_{k-1\|k-1}\boldsymbol{F}_{k-1}^T + \mathbf{Q}_{k-1}$	Prediction error covariance
Correction			
$\Delta\mathbf{z}_k$	=	$\tilde{\mathbf{z}}_k - h(\hat{\mathbf{x}}_{k\|k-1})$	Innovation
$\mathbf{S}_k$	=	$\mathbf{H}_k\mathbf{P}_{k\|k-1}\mathbf{H}_k^T + \mathbf{R}_k$	Innovation covariance
$\mathbf{K}_k$	=	$\mathbf{P}_{k\|k-1}\mathbf{H}_k^T\mathbf{S}_k^{-1}$	Kalman gain
$\hat{\mathbf{x}}_{k\|k}$	=	$\hat{\mathbf{x}}_{k\|k-1} + \mathbf{K}_k\Delta\mathbf{z}_k$	Estimated state
$\mathbf{P}_{k\|k}$	=	$(\mathbf{I} - \mathbf{K}_k\mathbf{H}_k)\mathbf{P}_{k\|k-1}$	Estimation error covariance

Table 2: Extended KF equations.

3.2 State evolution model

We consider a constant velocity model for the dynamics of the receiver,

$$\begin{cases} \mathbf{p}_k & = & \mathbf{p}_{k-1} + \mathbf{v}_{k-1} \cdot \Delta T \\ \mathbf{v}_k & = & \mathbf{v}_{k-1} \\ \mathbf{b}_{u,k} & = & \mathbf{b}_{u,k-1} + \mathbf{d}_{u,k-1} \cdot \Delta T \\ \mathbf{d}_{u,k} & = & \mathbf{d}_{u,k-1} \end{cases} \tag{15}$$

Exercice 8: WLS estimator for the dynamic receiver

1. In the main script **main_gnss.m**, line 22, set **Flag_spp** to 0. **Flag_propag** and **Flag_kml** are still set to 1.
2. In the script **user_config.m**, line 21, verify that **config.enable_EKF** is set to 0 (to use the WLS).
3. Run **main_gnss.m**. **What do you observe?**

   To help you, you can check the reference and estimated trajectories via Google Earth (put the buildings in 3D if possible), especially in the area between buildings, as well as the position of the satellites in parallel with the estimated $C/N0$ in Matlab figure 3 (samples 1050 to 1450).

Exercice 9: EKF for the dynamic receiver

1. Compute the state evolution matrix $\mathbf{F}$ from equation (15).
2. Complete lines 81 to 83 in **navigator/gnss_nav_ekf.m** (filter prediction).
3. In the script **user_config.m**:
   - rename the kml file as *navsol_trajectory_ekf* (variable **config.kml_file**, line 6),
   - set **config.kml_color**, line 9, to g (green),
   - set **config.enable_EKF** to 1.
4. Run the **main_gnss.m**. Which difference do you observe w.r.t. the WLS solution? **Comment on this point**.