Precise Point Positioning:
Centimeter accuracy trajectory reconstruction in peri-urban environment

Denis Laurichesse - April 2025
denis.laurichesse@gmail.com

Abstract

Achieving centimeter-level positioning accuracy in constrained environments remains a significant challenge for precise point positioning with ambiguity resolution (PPP-AR). This paper proposes a novel methodology that incorporates 10 Hz measurement processing, Doppler-aided vehicle dynamic modeling and precise ionosphere estimation, and N1 ambiguity gap bridging to address these challenges for moving receivers. The high-frequency processing enhances cycle slip detection and outlier rejection, ensuring robust preprocessing of GNSS measurements. Doppler observables are integrated into the filtering process to refine kinematic modeling and estimate ionospheric derivatives in real time, enabling rapid convergence even amid frequent signal obstructions. Additionally, a three-pass Kalman filter architecture bridges gaps in N1 ambiguities during outages, maintaining integer resolution without full reinitialization. Experimental results demonstrate that the proposed approach achieves centimeter-level horizontal accuracy in dynamic urban scenarios, outperforming conventional PPP-AR techniques. Using CNES GNSS corrections and multi-constellation support (GPS, Galileo, BeiDou), the methodology ensures compatibility across diverse satellite systems without the need of local corrections.

Keywords: GNSS · PPP · Ambiguity resolution

1. Introduction

Traditionally, precise Point Positioning (PPP) uses a single GNSS receiver along with precise orbit/clock corrections to achieve centimeter-level absolute accuracy in open-sky conditions. However, in challenging environments (like urban canyons), PPP performance degrades significantly – from centimeter-level accuracy in open-sky to decimeter-level or sub-meter accuracy in urban environments with heavy signal obstruction. These degradations result from reduced satellite visibility, multipath reflections (off buildings or vegetation), and frequent signal outages. Recent studies have investigated improvements to PPP, including ambiguity resolution (PPP-AR), multi-constellation (GPS, GLONASS, Galileo, BeiDou…), multi-frequency approaches, additional corrections (e.g., atmospheric, phase biases), and sensor integration, to maintain high accuracy in such challenging environments.

In dense urban environments, conventional PPP suffers from a long convergence time and degraded accuracy because of frequent satellite masking. Using multiple constellations improves geometry (PDOP) and increases satellite count, but this alone may not fully solve the problem. For example (Naciri, 2020) showed that resolving carrier-phase ambiguities (PPP-AR) drastically reduces convergence time and improves accuracy, to the extent that a dual-frequency GPS-only solution with AR converges faster and more accurately than a float solution (no AR) using GPS+GLONASS. Concretely, in datasets simulating an urban canyon (30° to 40° elevation mask), the PPP-AR solution reached convergence (<10 cm horizontal error) faster than the multi-constellation float solution and yielded ~50% lower horizontal errors1. Even with a 30° mask, the fixed (integer-resolved) solution converged more quickly than a float solution under open-sky conditions. These results underscore the value of PPP-AR: even with fewer satellites, fixing ambiguities can achieve ~1–2 cm horizontal accuracy, whereas a multi-GNSS float solution remains around 1.5–2 cm1. However, the vertical component remains more affected: in a 40° obstruction scenario, vertical RMS error reached ~9.5 cm, compared to ~1.7–1.9 cm in open-sky1, highlighting the difficulty of PPP in estimating altitude under strong signal blockage. Furthermore, after a certain point, adding more satellites (additional constellations) yields diminishing returns compared to enabling PPP-AR – in other words, resolving ambiguities on a few key constellations can provide a greater benefit than adding an extra constellation without AR. In dynamic urban settings (e.g., vehicles in motion), the added challenge is maintaining a fixed integer solution amidst frequent signal outages. Approaches like PPP-RTK (regional state-space corrections) combined with multi-GNSS and multi-frequency tracking have shown promise.

(Li, 2022) developed a triple-frequency multi-GNSS PPP-RTK model with regional iono/tropo corrections, tested in vehicular urban scenarios. Results: in suburban (less obstructed) areas, they achieved ~2 cm horizontal and 4 cm vertical accuracy, with ~94% of ambiguities correctly fixed1. Compared to GPS alone, accuracy improved by ~87%, largely thanks to Galileo, BeiDou, etc. In more obstructed urban settings with frequent signal breaks, their solution was able to re-fix ambiguities in ~5 seconds following each outage, maintaining centimeter-level tracking. These performances demonstrate that a fast multi-GNSS PPP-AR approach can support high-precision navigation in urban areas, provided access to external corrections (network-based) to speed up convergence. For standalone solutions without local networks, newly operational high-accuracy PPP services integrated with GNSS constellations provide possibilities.

In 2023, the Galileo High Accuracy Service (HAS) began broadcasting free PPP corrections (orbits, clocks, code biases) on E6. (Yi, 2023) tested Galileo HAS in an urban driving scenario with a low-cost dual-frequency receiver. Because HAS does not yet provide phase biases (thus only enabling PPP float), the GPS+Galileo PPP (HAS) solution achieved accuracy comparable to solutions using CNES ultra-rapid products (PPP-WIZARD project,) at the 68th percentile. In practice, this implies that the horizontal error was mostly decimeter-level, similar to a reference PPP solution. The study also reported that adding a low-cost inertial measurement unit (IMU) in tight coupling significantly enhanced robustness: overall RMS error improved by ~70% compared to GNSS-only PPP. However, IMU integration did not significantly reduce median (68%) errors, which are primarily driven by the GNSS component already being filtered. Overall, in urban environments, a multi-GNSS PPP-AR solution can achieve ~0.3–0.5 m real-time horizontal accuracy (float) on a moving vehicle, and further approach decimeter-level accuracy by coupling with an IMU.

Environment	PPP Method	Constellations	Achieved Accuracy	Conditions / Notes	Source
Open-sky (reference)	PPP-AR (fixed ambiguities, static)	GPS + Galileo (+ others)	~1–2 cm horiz. ~2–5 cm vert.	Geodetic station, >30 min for full convergence.	(Naciri, 2020)
Urban canyon (static)	PPP float (no integer fixing)	GPS+GLONASS+ Gal+BDS	~1.5–2.4 cm horiz. (RMS) ~2–9 cm vert.	30–40° mask. Accuracy varies with obstruction: horizontal remains cm-level, vertical may degrade to ~9 cm.	(Naciri, 2020)
Urban canyon (static)	PPP-AR, dual-frequency	GPS	~0.9 cm horiz.	Same site as above. AR shortens convergence (~3.5 min faster vs GPS+GLONASS float) and reduces error by ~50%.	(Naciri, 2020)
Urban canyon (vehicle)	Real-time PPP-RTK (ambiguities fixed)	Multi-GNSS (GPS+Gal+BDS+ GLO)	~2 cm horiz. ~4 cm vert.	Suburban driving test (less obstructed). 93.7% fix rate. Ambiguities refixed in ~5 s after outages.	(Li, 2022)

Environment	PPP Method	Constellations	Achieved Accuracy	Conditions / Notes	Source
Urban canyon (vehicle)	Real-time PPP float (HAS SSR)	GPS+Galileo	~0.3–0.5 m horiz. (68%)	Dense urban scenario. Galileo HAS service (no phase bias → float PPP). 68% accuracy on par with reference PPP. Adding IMU → ~70% RMS improvement.	(Yi, 2023)

Table 1: Performance Comparison of PPP by Environment & Method

To the best of our knowledge, there is no centimeter-level positioning result for a moving receiver, using the PPP-AR technique without external assistance such as ionosphere or troposphere requiring a local network of stations, in constrained environments. In this paper, we propose a solution to this problem by adding some key elements to the classic resolution techniques.

2. Overview of the process of trajectory computation

The diagram below illustrates the complete process for computing the trajectory:

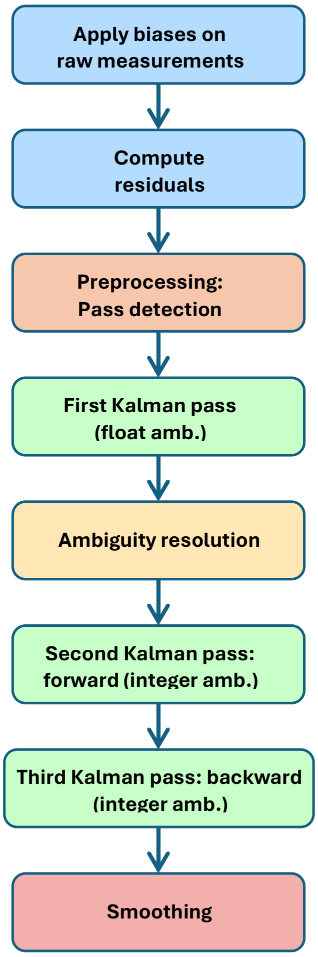
Apply biases on raw measurements
Compute residuals
Preprocessing: Pass detection
First Kalman pass (float amb.)
Ambiguity resolution
Second Kalman pass: forward (integer amb.)
Third Kalman pass: backward (integer amb.)
Smoothing

Figure 1: Trajectory estimation data flow

3. Choice of frequency and Observables

(Laurichesse D., 2018) showed that in a three-frequency context, the optimal frequency plan to minimize measurement noise in a PPP-AR context consists of selecting the two most widely spaced frequencies, and as the third frequency, the one that is most 'centered' between the other two. This leads to the frequency choices circled in red in the following figure.

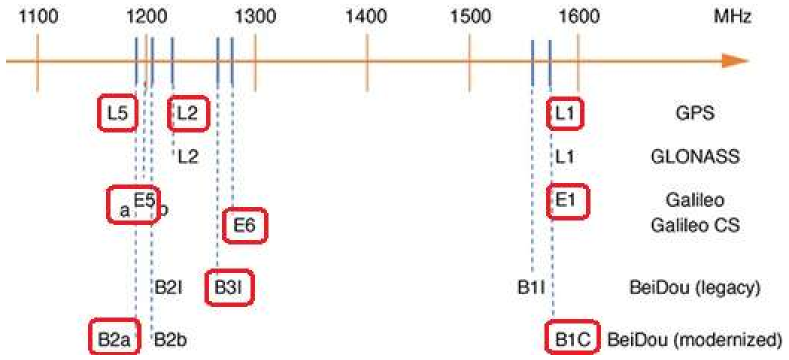

Figure 2: GNSS frequencies and selection (source GPS World)

After frequency selection, we perform a combination of individual phase measurements according to the following table:

Constellation	Extra-widelane	Widelane	N1	Noise
GPS	L5 – L2	L2 – L1	L1	0.05 cycle
Galileo	L6 – L5	L5 – L1	L1	0.05 cycle
Beidou	L5 – L3	L3 – L1	L1	0.05 cycle

Table 2: Choice of phase combinations

Here is the choice of the frequencies for the code measurements:

Constellation	Frequency a	Frequency b	Noise
GPS	P1	P2	1 m
Galileo	P1	P5	1 m
Beidou	P1	D5	1 m

Table 3: Choice of code measurements

Here is the choice of the frequencies for the Doppler measurements:

Constellation	Frequency a	Frequency b	Noise
GPS	D1	D2	0.15 m/s
Galileo	D1	D5	0.15 m/s
Beidou	D1	D5	0.15 m/s

Table 4: Choice of Doppler measurements

4. CNES products

The **PPP-WIZARD** project (Gazzino, 2023) (PPP-WIZARD project,) by CNES (Centre National d'Études Spatiales) provides post-processed GNSS correction products aimed at enhancing positioning accuracy for users performing precise point positioning (PPP) in post-processing mode. These products are particularly useful for testing PPP algorithms under conditions that simulate real-time data processing, while benefiting from the precision of post-processing.

The key features of PPP-WIZARD Post-Processed Corrections are the following:

- **Daily Correction Files:** The project offers daily files containing satellite orbit and clock corrections. These files are essential for users to test their PPP software in post-processing while simulating real-time conditions.
- **High-Precision Data:** The corrections include precise satellite orbits and clocks, crucial for achieving high-accuracy positioning results in post-processed PPP applications.
- **Compatibility with Open-Source Tools:** The format of these files is standardized by the IGS (Johnston, 2017).
- **Support for Ambiguity Resolution:** The system supports zero-difference ambiguity resolution methods, enhancing the reliability and accuracy of PPP solutions
- **Multi-Constellation and Multi-Frequency Support:** PPP-WIZARD products support multiple GNSS constellations, including GPS, GLONASS, Galileo, and BeiDou, as well as multiple frequencies per constellation. This multi-constellation capability improves the reliability and accuracy of PPP solutions, especially in challenging environments where satellite visibility may be limited.
- **Code and phases biases are available in a BIA format** (Sinex BIA FORMAT,). It is required to unbias raw measurements before proceeding to residual computation and ambiguity resolution.

5. Measurement Model

Measurements residuals are computed using RTKLIB (RTKLIB library,). The initial solution is the receiver PVT, a convenient choice to linearize the filter equations. The following table summarizes the terms used to model the measurements.

Model term	Approx. amplitude (m)
Propagation distance	10^7
Attitude law	1
Troposphere	0.50
Solid earth tide	0.20
Phase Wind-Up	0.20
Satellites and stations PCV and PCO	0.10
Relativistic effects	0.01

Table 5: Characteristics of the measurement model

6. Preprocessing

The principle of the preprocessing step is to detect large jumps or inconsistencies in the phase or code measurements by looking at the temporal differences of linear combinations. For robustness, if an outlier is detected on one combination, all the measurements are deemed invalid for this satellite. It is shown that, at 10 Hz, this process is sufficient to eliminate all the cycle slips of the phase and most outliers for the code. No preprocessing is applied for the Doppler measurements. Correct cycle slip detection is essential for a proper separation of ambiguities. In addition, an outlier rejection scheme is implemented on the main filter.
The following tables show the various combinations and thresholds for the preprocessing step.

Combinations	Phase a	Phase b	Threshold
GPS	λ_5*L_5 – λ_1*L_1	λ_2*L_2 – λ_1*L_1	0.05 cm
Galileo	λ_6*L_6 – λ_1*L_1	λ_5*L_5 – λ_1*L_1	0.05 cm
Beidou	λ_5*L_5 – λ_1*L_1	λ_3*L_3 – λ_1*L_1	0.05 cm

Table 6: Phase combinations for preprocessing

Combinations	Code a	Code b	Threshold
GPS	P_5 – P_1	P_2 – P_1	2 m
Galileo	P_6 – P_1	P_5 – P_1	2 m
Beidou	P_5 – P_1	P_3 – P_1	2 m

Table 7: Code combinations for preprocessing

Note that such stringent thresholds may lead to an increased number of cycle slip detections and thus can increase the number of passes in the filter. There are typically 3000 passes in a 20-minute run.

7. First pass: forward Kalman filter with float ambiguities

In this section, we review the formulation of the observation equations for pseudorange, carrier phase, and Doppler measurements (Laurichesse D. &., 2015). The equations are formulated for the GPS case but can be easily extended to Galileo and Beidou. We will use the following constants in equations (k = 1,2 or 5):

$$\begin{aligned} \gamma_k &= \frac{f_1^2}{f_k^2} \\ \lambda_k &= \frac{c}{f_k} \end{aligned}$$

Code and Phase Measurements:
The observation equations for code (P/C) and phase (L) are:

$$\begin{aligned} P_1 + \Delta b_{P_1} &= \rho_1 + e + \Delta\tau \\ P_2 + \Delta b_{P_2} &= \rho_2 + \gamma_2 e + \Delta\tau \\ C_5 + \Delta b_{C_5} &= \rho_5 + \gamma_5 e + \Delta\tau \\ \lambda_1\left(L_1 + \Delta b_{L_1}\right) &= \rho_1 + \lambda_1 W - e + \Delta\tau - \lambda_1 N_1 \\ \lambda_2\left(L_2 + \Delta b_{L_2}\right) &= \rho_2 + \lambda_2 W - \gamma_2 e + \Delta\tau - \lambda_2(N_W + N_1) \\ \lambda_5\left(L_5 + \Delta b_{L_5}\right) &= \rho_5 + \lambda_5 W - \gamma_5 e + \Delta\tau - \lambda_5(N_E + N_W + N_1) \end{aligned}$$

Doppler Measurements

The Doppler measurement (D_k, in Hz) is related to the rate of change of the phase measurement. The functional model, expressed in meters per second ($\lambda_k D_k$), is:

$$\begin{aligned} \lambda_1 D_1 &= \dot{\rho}_1 - \dot{e} + \Delta\dot{\tau} \\ \lambda_2 D_2 &= \dot{\rho}_2 - \gamma_2 \dot{e} + \Delta\dot{\tau} \\ \lambda_5 D_5 &= \dot{\rho}_5 - \gamma_5 \dot{e} + \Delta\dot{\tau} \end{aligned}$$

Where:

- ρ_k, are the geometrical propagation distances between the emitter and receiver phase centers, including all effects detailed in the measurement model paragraph.
- $\dot{\rho}_k$ is the geometric range rate (relative velocity component along the line-of-sight) for frequency k.
- W is the contribution of the wind-up effect (in cycles).
- e is the ionosphere elongation in meters at f_1.
- $\dot{e}$ is the rate of change of the ionospheric elongation at f_1 (in meters per second).
- $\Delta\tau = \tau_i - \tau^j$ is the difference between receiver i and emitter j clocks.
- $\Delta\dot{\tau} = \dot{\tau}_i - \dot{\tau}^j$ is the relative clock drift (difference between receiver and satellite clock drifts) in meters per second.
- N_1, N_W, and N_E are the integer carrier phase ambiguities, respectively N1, Widelane, and Extra-widelane.
- $\Delta b_p = b_{p_i} - {b_p}^j$ is the difference between receiver i and emitter j code biases.
- $\Delta b_L = b_{Li} - {b_L}^j$ is the difference between receiver i and emitter j phase biases.

The combination $\tau_i + b_{pi}$ (or $\tau_i + b_{Li}$) for each observable type represents a receiver clock term. This approach is equivalent to the decoupled clock model, where one distinct receiver clock parameter is estimated for each signal type (e.g., P1, P2, C5, L1, L2, L5). For a detailed discussion on decoupled clock models, see (Collins, 2010).

For other constellations like Galileo and BeiDou, the functional model follows the same principle. The primary difference lies in the specific frequencies (f_k) used, which determines the corresponding wavelengths (λ_k) and frequency-dependent factors (γ_k). The exact frequency plans can be found in the official Signal-In-Space Interface Control Documents (SIS ICDs) published for each constellation.

Filter formulation

The filtering uses a Kalman filter in square root covariance formulation. Propagation and update steps are performed using QR transformations. The general formulation of the Kalman filter is of the $P = LL'$ form, with L an lower triangular matrix.

The filter follows the functional model, treating satellite positions, clock corrections, and biases as fixed inputs rather than parameters to be estimated.

State Propagation

The state vector x (correction to the pvt, the residuals initial reference) is updated as follows:

$$\boxed{\begin{gathered} x_{pvt} \leftarrow x_{pvt} + pvt_{t-1} \\ x \leftarrow \Phi \cdot x \\ x_{pvt} \leftarrow x_{pvt} - pvt_t \end{gathered}}$$

t being the current epoch, and t-1 the previous epoch. The transition matrix Φ is essentially the identity except for the positions/velocities and iono/iono derivative coupling:

$$\Phi_{P/V} = \begin{pmatrix} \mathrm{I} & \Delta T \\ 0 & I \end{pmatrix}, \text{ and } \Phi_{iono/\dot{\iota ono}} = \begin{pmatrix} \mathrm{I} & \Delta T \\ 0 & I \end{pmatrix}$$

ΔT being the time increment between two successive steps (typically 0.1 sec).

In the Kalman filter, the off-diagonal elements of the state transition matrix introduce cross-dependencies between state variables, preventing independent optimization and complicating computations due to increased coupling. This makes the system harder to simplify and computationally more demanding.

Φ is arranged to have r non diagonal elements as first parameters and diagonal elements (typically phase ambiguities and clocks) at the end. One can write:

$$\Phi = \begin{pmatrix} \Phi_r & 0 \\ 0 & I \end{pmatrix} \text{ and } L = \begin{pmatrix} L_r \\ L_e \end{pmatrix}$$

Covariance Propagation

The square-root covariance matrix L is propagated as follows. First, compute:

$$\Phi_L = \begin{bmatrix} \Phi_r \cdot L_r \\ L_e \end{bmatrix}$$

Construct the matrix for QR decomposition:

$$M = \begin{bmatrix} {\Phi_L}^{\mathrm{T}} \\ \Sigma_{\mathrm{Mod}} \end{bmatrix}$$

With Σ_{Mod} the model noise covariance matrix. Then, we apply a QR decomposition:

$$\boxed{\mathrm{QR} = \mathrm{qr}(M) \text{ and } L \leftarrow R^{\mathrm{T}}}$$

Correction Step

With the following notations:

H : observation matrix.
B : measurement vector.
S : standard deviation vector of measurement noise.
r : innovation vector (residual).

We begin by computing the measurement residual:

$$r = B - Hx$$

We build the matrix M used in the QR decomposition:

$$M = \begin{bmatrix} HL & -\mathrm{diag}(S) \\ L & 0 \end{bmatrix}^{\mathrm{T}}$$

This matrix stacks the measurement model and the prior uncertainty in a way that prepares it for the orthogonal transformation step. We apply a QR decomposition:

$$\mathrm{QR} = \mathrm{qr}(M)$$

We transpose the result to extract L:

$$L \leftarrow R^{\mathrm{T}}$$

We extract the lower-triangular matrix R_e, the gain-like matrix G_k and the updated covariance matrix L:

$$\mathrm{Z} \ = \begin{bmatrix} R_e & 0 \\ G_k & L \end{bmatrix}$$

The triangular matrix R_e serves as the post-fit residual covariance square-root, while G_k acts as a gain matrix. We compute the updated state vector:

$$\boxed{x \leftarrow x + G_k R_e^{-1} r}$$

This expression performs the square-root equivalent of the standard Kalman update equation.

Finally, the residual can be recomputed using the updated state:

$$r = B - Hx$$

This updated residual vector is used for quality checks. If at least one component of r is above 3*S, there is an outlier. We take the outlier with the maximum value, remove it from the measurements, and repeat iteratively the correction process.

Parameterization and tuning of the filter

The following table summarizes the characteristics of the filter:

Parameter	Initial covariance	Model noise (10 Hz step)
Positions (3)	1000 m	0
Velocities (3)	100 m/s	0.1 m/s
ZTD	0.1 m	10^{-5} m
GPS clocks - 2 Code - 3 Phase - 2 Doppler	 ∞ ∞ ∞	 ∞ ∞ ∞
GPS iono (1 per sat)	1000 m	0
GPS iono derivative (1 per sat)	100 m/s	10^{-5} m/s
GPS EWL ambiguity (1 per pass)	1000 cycle	0
GPS WL ambiguity (1 per pass)	1000 cycle	0
GPS L1 ambiguity (1 per pass)	1000 cycle	0
GALILEO clocks - 2 Code - 3 Phase - 2 Doppler	 ∞ ∞ ∞	 ∞ ∞ ∞
GALILEO iono (1 per sat)	1000 m	0
GALILEO iono derivative (1 per sat)	100 m/s	10^{-5} m/s
GALILEO EWL ambiguity (1 per pass)	1000 cycle	0
GALILEO WL ambiguity (1 per pass)	1000 cycle	0
GALILEO L1 ambiguity (1 per pass)	1000 cycle	0
Beidou clocks - 2 Code - 3 Phase - 2 Doppler	 ∞ ∞ ∞	 ∞ ∞ ∞
Beidou iono (1 per sat)	1000 m	0
Beidou iono derivative (1 per sat)	100 m/s	10^{-5} m/s
Beidou EWL ambiguity (1 per pass)	1000 cycle	0
Beidou WL ambiguity (1 per pass)	1000 cycle	0
Beidou L1 ambiguity (1 per pass)	1000 cycle	0

Table 8: Characteristics of the Kalman filter

8. Integer ambiguity Resolution

Ambiguity Datum

In (Collins, 2010), the "ambiguity datum" refers to the minimal set of integer ambiguities that are fixed to anchor the phase clock parameters. Because code and phase clock parameters are estimated separately (decoupled), the system becomes rank-deficient unless some reference is established for the carrier-phase. By fixing these datum ambiguities, essentially selecting certain integer ambiguities as a baseline, the solution removes this rank deficiency. It ensures that all other ambiguities in the network or single-station solution remain strictly integer-valued and can be resolved accordingly. This step is mandatory when the ambiguities are expressed in undifferenced formulation.

Bootstrap

In GNSS precise point positioning with ambiguity resolution (PPP-AR), the bootstrap method is a simple, sequential approach to resolving integer ambiguities. First, the float solution is computed, producing real-valued estimates for each carrier-phase ambiguity. The bootstrap algorithm then fixes one integer ambiguity at a time, usually starting with the one that has the smallest formal uncertainty, by rounding its float estimate to the nearest integer. After fixing that ambiguity, the process is repeated for the next best-conditioned ambiguity, incorporating the newly fixed integer value to refine subsequent estimates. Although less optimal than advanced techniques like LAMBDA in terms of success rates and robustness, the bootstrap method is straightforward and can still deliver correct integer solutions under favorable conditions.

The bootstrap method requires a floating-point solution and its associated covariance matrix as input. The solution may consist of either independent ambiguities or combinations thereof. In our approach, we use the independent ambiguity solution to resolve extra-wide-lane and wide-lane ambiguities. Subsequently, we compute all linear combinations of N1 ambiguities across successive satellite passes, enabling us to perform "gap bridging" ambiguity resolution. The goal of gap-bridging is to connect two successive passes to lower the covariances of their ambiguities.

Gap bridging

We compute all the differences for ambiguities belonging to two successive satellite passes. This gives matrix A that transforms the original state of ambiguities:

$$\mathbf{A} = \begin{pmatrix} \cdots & -1 & \cdots & 1 \\ & \cdots & & \end{pmatrix}$$ (the -1 and 1are located at the indexes of successive N1 passes)

Then we form the transformed state vector:

$$\mathrm{y} = \mathrm{Ax}$$

And the associated covariance matrix:

$$\mathrm{Q}_y = \mathrm{ALL}^T\mathrm{A}^T$$

A bootstrap resolution is then applied on $(y \quad \mathrm{Q}_y)$. The solution of this bootstrap is introduced in the main filter as a set of measurements (with design matrix A).

Cascading process

We solve for extra-widelanes and widelanes, we bridge N1 gaps and then we resolve the N1 ambiguities. Finally, we output two integer ambiguity solutions: the first includes all ambiguity resolutions except the final N1 ambiguities (**solution A**), while the second solution incorporates all ambiguities fully resolved (**solution B**).
This cascading process is summarized in the following diagram:

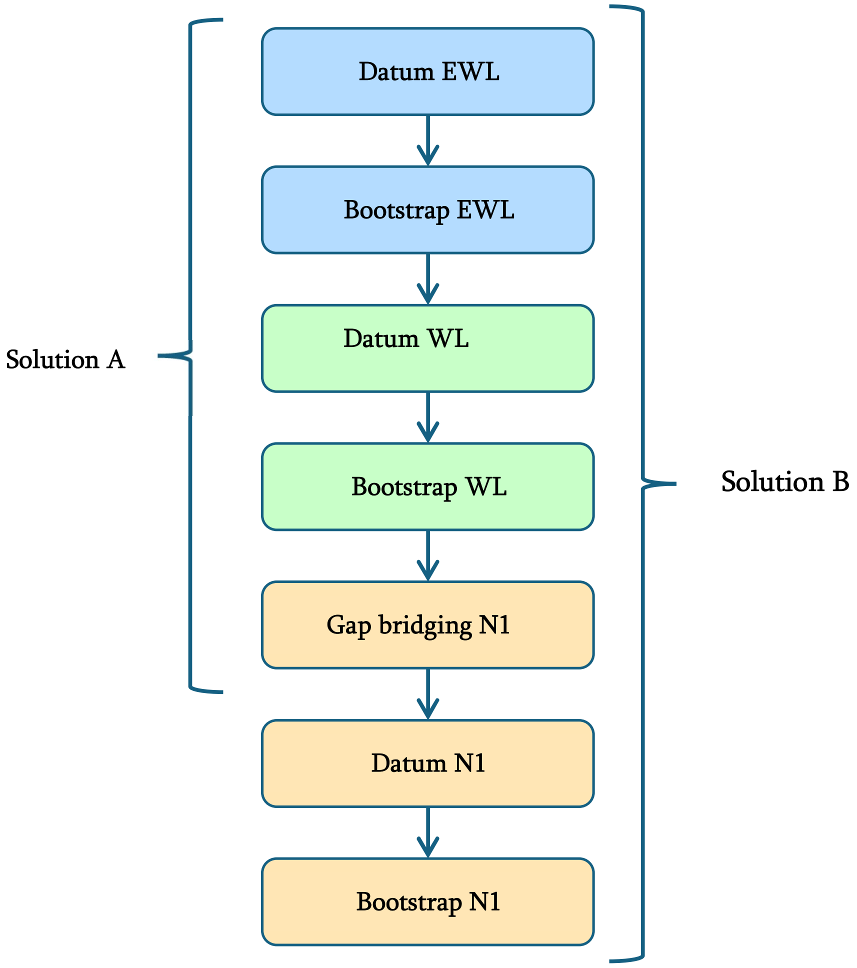

Figure 3: Ambiguity resolution data flow

9. Second pass: forward Kalman filter with fixed ambiguities (unsolved ambiguities remain as float parameters)

The final forward solution is computed using the same Kalman filter. The difference is that the initial state and covariance matrix are adjusted for already resolved individual ambiguities or combinations thereof. This gives the $(\mathbf{x}_f, \mathbf{P}_f)$ solution.

10. Third pass: backward Kalman filter

The backward solution uses the same principles and data as the forward one (second pass), but the epochs are processed in reverse. This gives the $(\mathbf{x}_b, \mathbf{P}_b)$ solution.

11. Combination of the solutions

We use the forward and backward state and covariance, respectively: $(\mathbf{x}_f, \mathbf{P}_f)$ and $(\mathbf{x}_b, \mathbf{P}_b)$, then, at each epoch t, we compute the combined solution:

The combined covariance matrix $\mathbf{P}_t$ is computed as the inverse of the sum of the information matrices (inverse covariances) from the forward and backward passes:

$$\mathbf{P}_t = \left(\mathbf{P}_{f,t}^{-1} + \mathbf{P}_{b,t}^{-1}\right)^{-1}$$

The combined state estimate $\mathbf{x}_t$ is a weighted average of the forward and backward estimates, where each estimate is weighted by its respective information matrix. The result is then scaled by the combined covariance matrix:

$$\mathbf{x}_t = \mathbf{P}_t\left(\mathbf{P}_{f,t}^{-1}\mathbf{x}_{f,t} + \mathbf{P}_{b,t}^{-1}\mathbf{x}_{b,t}\right)$$

12. Real life scenarios

Receiver & reference trajectory reconstruction

The receiver used to collect GNSS measurements is a Septentrio AsteRx-U associated with a Zephyr 2 antenna.

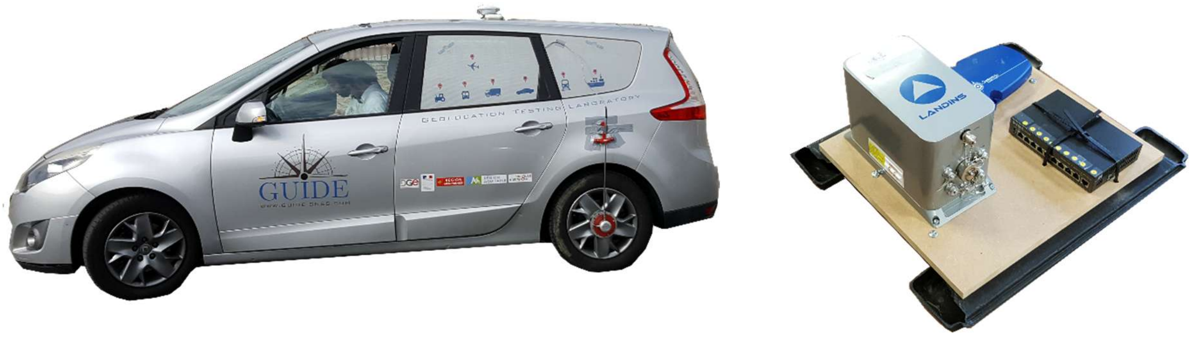

Figure 4: Test Vehicle Equipped with reference sensors (left), instrumented measurement platform (right), courtesy Guide GNSS (Guide GNSS,)

To compute a high-precision GNSS reference trajectory, a multisensor platform was deployed on a test vehicle (see figure). The setup includes a LANDINS inertial measurement unit (IMU), an AsteRx-U GNSS receiver paired with a Zephyr 2 antenna, and a calibrated wheel odometer mounted on the vehicle's rear wheel.

Additionally, corrections from reference GNSS stations were applied to enhance positioning accuracy. The raw sensor data were recorded during test drives and subsequently processed using specialized post-processing software capable of loosely coupling GNSS and inertial measurements.

This loose coupling enables robust trajectory reconstruction even in partially obstructed environments. The achieved absolute positioning accuracy is within ±3 cm, as guaranteed by the post-processing solution, and forms the cornerstone of the reference trajectory's reliability.

Test run #1

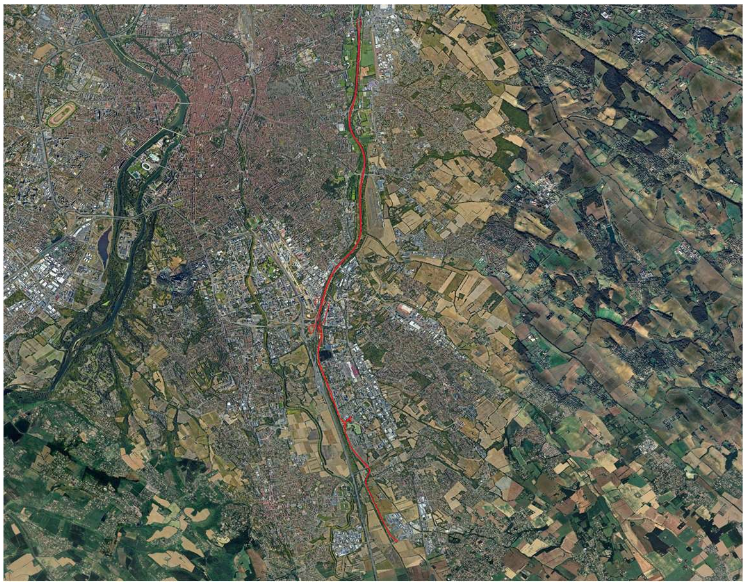

Figure 5: Maps of trajectory #1, (courtesy Google maps)

The route lasts about 45 minutes (on 11/7/2024) and mostly follows the ring road around Toulouse, France.

- Mixed semi-urban zones with moderate building density, industrial zones, and some open spaces. Reduced multipath compared to the city center, but still susceptible to partial signal attenuation or reflection.
- Peri-urban or rural areas, with open fields, sparse buildings, and agricultural lands. Generally good signal reception, minimal multipath or obstruction.

The trajectory is mostly linear, with moderate to high-speed travel with minimal curves — suitable for dynamic GNSS analysis.

The following figure shows the horizontal errors for the receiver's PVT:

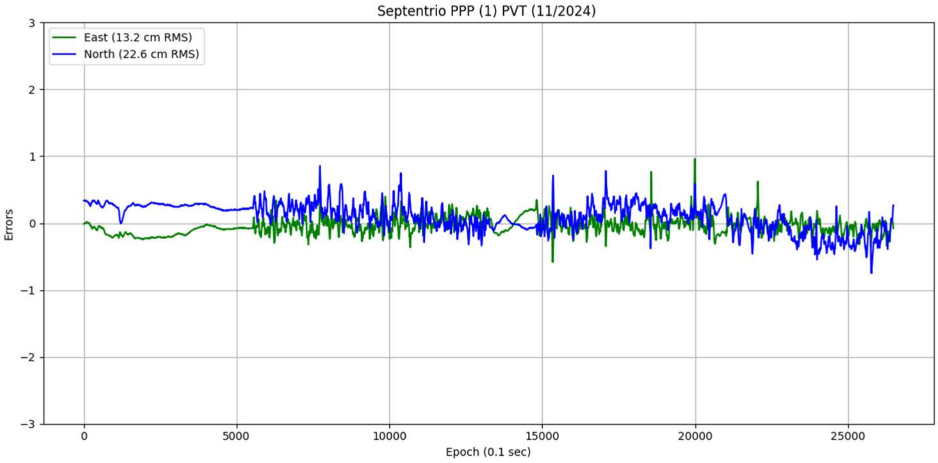

Figure 6: receiver PVT errors

The PVT trajectory is relatively smooth, without outliers above 1 m.

The following figure shows the horizontal errors of the filtered trajectory (solution A):

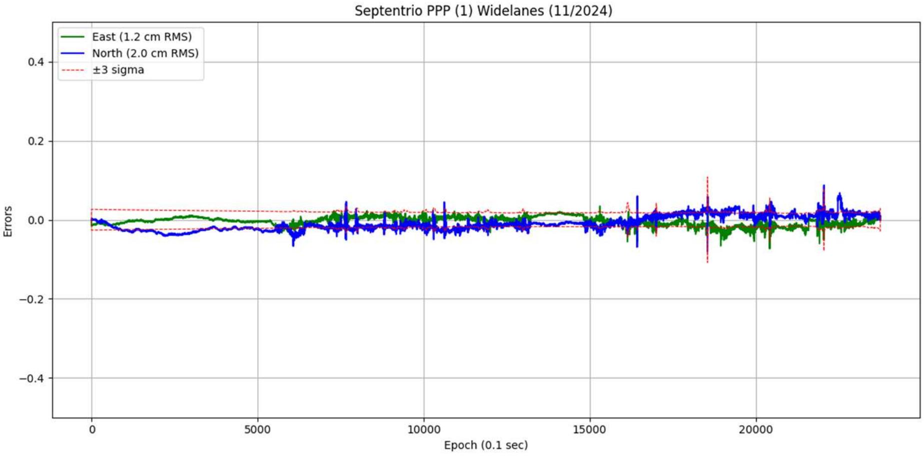

Figure 7: solution A errors (for stats, epochs where sigma > 3 cm are removed)

We notice the high accuracy of the obtained trajectory, even with partial ambiguity fixing.

The following figure shows the horizontal errors of the filtered trajectory (solution B):

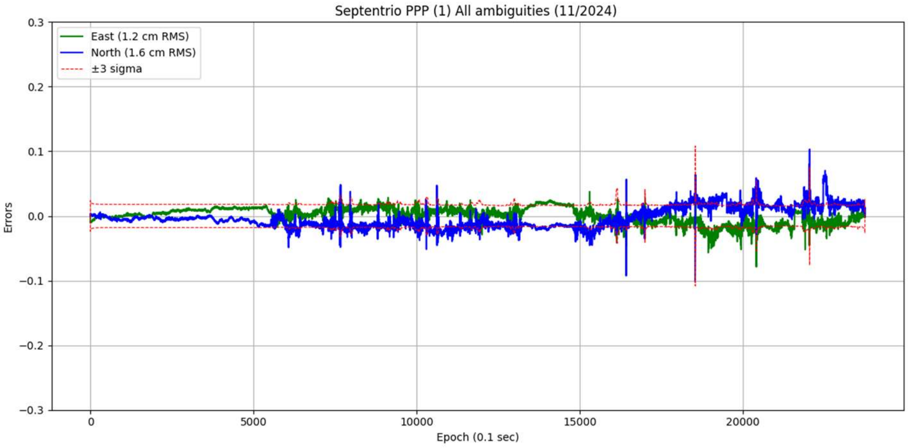

Figure 8: solution B errors (for stats, epochs where sigma > 3 cm are removed)

With all ambiguities fixed, the trajectory has a better accuracy. Yet, the gain is marginal compared to solution A.

Test run #2

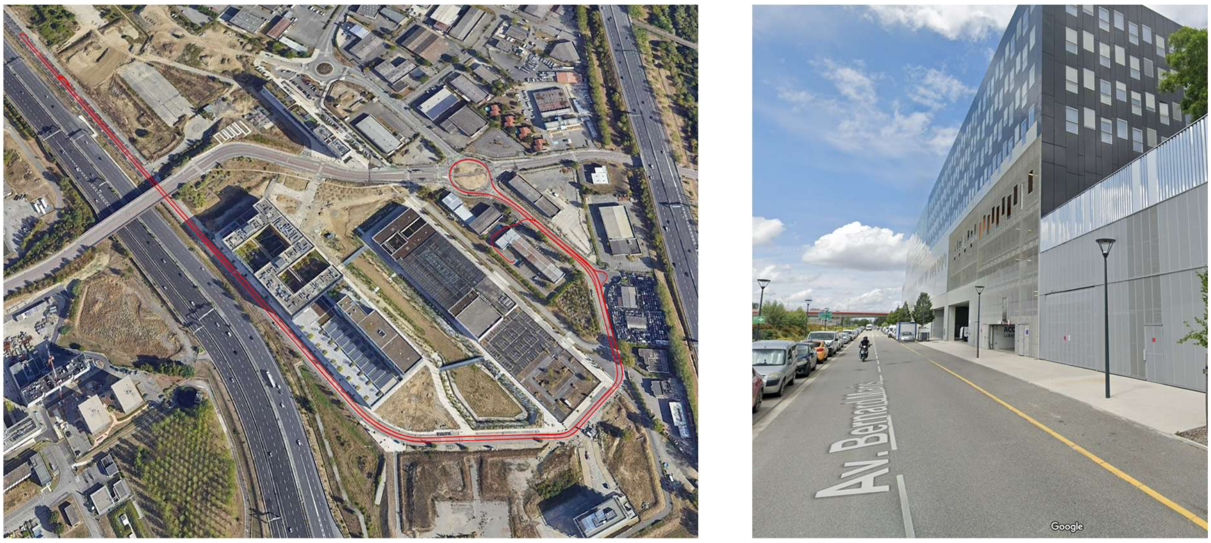

Figure 9: Maps of trajectory #2 (left), zoom on the multipath zone (right), (courtesy Google maps)

The vehicle is navigating through a semi-urban industrial area, featuring multiple buildings, parking areas, and road infrastructure such as overpasses and junctions.

The main feature of this test is the passage along a building, which causes masking and multipath effects.

- The tall building on the right side creates a major masking zone, especially for satellites that would be low on the sky in the north directions.
- This reduces the number of visible satellites, which may degrade positioning geometry (higher GDOP) and solution reliability.
- The glass and metallic surfaces of the building are highly reflective and will cause GNSS signals to bounce.
- These multipath signals introduce biases and noise into both pseudorange and carrier-phase measurements.

The following figure shows the horizontal errors for the receiver's PVT:

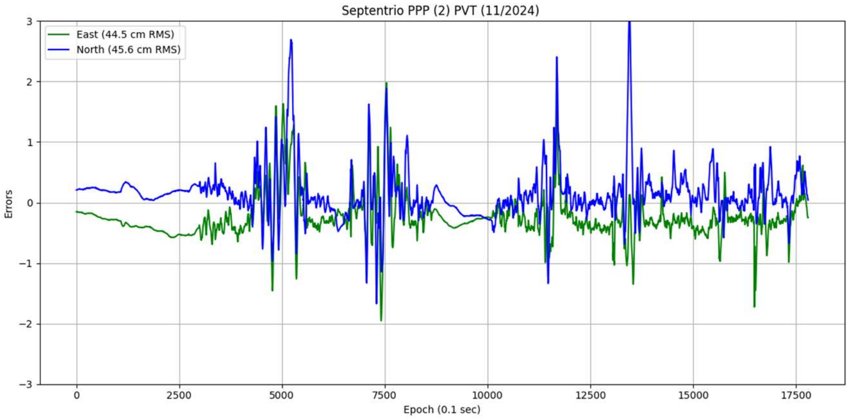

Figure 10: receiver PVT errors

Here, we can observe that the trajectory is noisier than in run #1, mainly due to the repeated passes along the building and under the bridge.

The following figure shows the horizontal errors of the filtered trajectory (solution A):

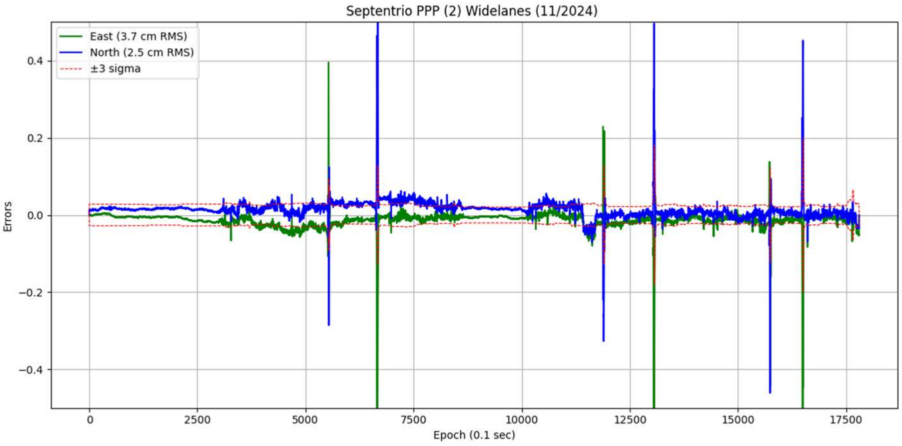

Figure 11: Solution A errors (for stats, epochs where sigma > 3 cm are removed)

On this trajectory, bridge underpasses are clearly visible. There is a sudden drop in the number of measurements, resulting in a sharp increase in formal covariance. However, we note that phase continuity could be maintained along the building.

This shows that, even when performing only partial ambiguity resolution, one can perform below 5 cm accuracy.

The following figure shows the horizontal errors of the filtered trajectory (solution B):

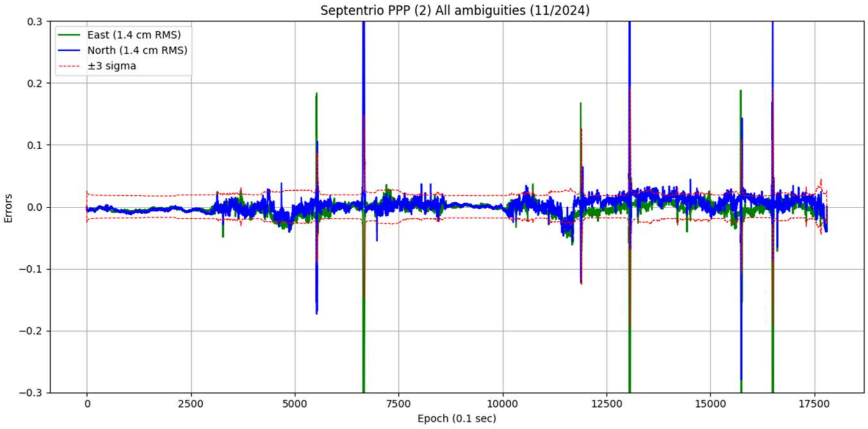

Figure 12: Solution B errors (for stats, epochs where sigma > 3 cm are removed)

As in run #1, there is an improvement over solution A. This comes at the cost of successfully resolving the N1 ambiguities, which can be a challenging task, particularly when it comes to their validation.

13. Synthesis of the solutions

	Run #1	Run #2
Run duration	40 min	29 min
# epochs (10 Hz)	23750	17800
# passes	2235	1864
# parameters	2519	2148
EWL amb. fixing rate	98 %	99 %
WL amb. fixing rate	95 %	97 %
Gap Bridging N1 fixing rate	80 %	70 %
N1 amb. fixing rate	87 %	63 %
PVT 2D error	26 cm	64 cm
PPP (A) 2D error / avail	2.3 cm / 100 %	4.4 cm / 99.4 %
PPP (B) 2D error /avail	2 cm / 99.9 %	1.9 cm / 98.9 %

Table 9: Synthesis of run's main results

14. Challenges

This technique is promising, but few challenges remain:

- Ambiguity fixing may be instable. It is advised to implement a robust ambiguity validation technique (Banville, 2021).
- In some very complex environments (or in real-time scenarios), Solution B may not be reached. Solution A is the preferred option (with a limited degradation of performance).
    - *Important note: On the server side (GNSS product generation), the station network required for computing Solution A (widelane biases) is estimated at approximately 50 stations, which is significantly fewer than the ~150 stations required for computing Solution B (full-phase bias estimation).*
- The process is not compatible with real-time execution, but there are no theoretical limitations to adapting it for real-time purposes.
- The processing time is currently very long due to the high number of passes and the large state vector. Techniques may be applied to reduce the number of passes or to accelerate the computational load. For example, GPUs can be used to accelerate the QR transformations. Below is a brief benchmark of a typical QR transformation across various configurations:

Processor configuration (google colab)	Time (1 QR transform)
CPU (2 threads)	800 ms
GPU T4	200 ms
GPU L4	80 ms
GPU A100	20 ms

Table 10: Computation time of various configurations

15. Conclusion

This paper addresses the challenge of achieving PPP-AR in constrained environments by introducing a methodology that integrates **high-frequency (10 Hz) processing**, **Doppler-aided dynamic modeling accurate ionosphere estimation**, and **N1 ambiguity gap bridging**. These elements enable centimeter-level positioning for moving receivers in peri-urban conditions, without reliance on external atmospheric corrections or local reference networks.

Acknowledgments

The author wishes to thank Simon Banville for his helpful comments and the GUIDE company (Guide GNSS,) for providing the data sets (Rinex files and reference trajectories).

References

1. **PPP-WIZARD project**: Retrieved from [www.ppp-wizard.net]
2. **RTKLIB library**: Retrieved from [www.rtklib.com]
3. **Guide GNSS**: Retrieved from [https://guide-gnss.com].
4. **Banville, S. & al. (2021)**: Enabling ambiguity resolution in CSRS-PPP. Navigation: Journal of The Institute of Navigation. doi(https://doi.org/10.1002/navi.42)
5. **Collins, P. B. (2010)**: Undifferenced GPS Ambiguity Resolution using the Decoupled Clock Model and Ambiguity Datum Fixing. Navigation: Journal of the Institute of Navigation, 123 - 135.
6. **Gazzino, C. & al. (2023)**: The CNES Solutions for Improving the Positioning Accuracy with Post-Processed Phase Biases, a Snapshot Mode, and High-Frequency Doppler Measurements Embedded in Recent Advances of the PPP-WIZARD Demonstrator. Remote Sensing.
7. **Johnston, G. R. (2017)**: The International GNSS Service. Springer International Publishing, 1st ed., pp. 967-982. doi(https://doi.org/10.1007/978-3-319-42928-1)
8. **Laurichesse D., Banville S. (2018)**: Innovation: Instantaneous Centimeter-Level Multi-Frequency Precise Point Positioning. GPS World.
9. **Laurichesse, D. & al. (2015)**: An Open-source PPP Client Implementation for the CNES PPP-WIZARD Demonstrator. Proceedings of the ION GNSS+. Tampa.
10. **Laurichesse, D. (2015)**: Innovation: Carrier-Phase Ambiguity Resolution. GPS World.
11. **Li, X. W. (2022)**: Principle and performance of multi-frequency and multi-GNSS PPP-RTK. Satellite Navigation. doi(https://doi.org/10.1186/s43020-022-00068-0)
12. **Naciri, N. & al. (2020, October)**: Multi-GNSS Ambiguity Resolution For Signal Obstruction in PPP. Inside GNSS.
13. **Sinex BIA FORMAT**: Retrieved from [http://www.aiub.unibe.ch/download/bcwg/format/draft/sinex_bias_100_dec07.pdf]
14. **Yi, D. & al. (2023)**: Galileo HAS: A Performance Assessment in Urban Driving Environment. Inside GNSS.