



NAVISP1-88

**PROOF-OF-CONCEPT OF
ADVANCED NAVIGATION
ALGORITHMS BASED ON
FACTOR GRAPH OPTIMIZATION
-
FINAL PRESENTATION HANDOUT**

05 October 2026

Objective of the activity

The objective of the NAVISP EL1-088 Project was to investigate innovative navigation algorithms exploiting Factor Graph Optimization (FGO) methods for PVT estimation, both standalone and in combination with other sensors, and for different receiver grades (including high- and low-quality observables).

This objective was achieved via the design and implementation of a software-based FGO navigation (in brief FGO-NAV) engine that is integrated in a flexible testbed composed of three GNSS receivers having integrated IMU sensors (two low-end and a high-end receiver).

Our activity benchmarked both the potential accuracy improvement and the computational cost of the proposed algorithms.

On top of these objectives proposed by the ESA, the ABBIA's project team added another objective: explore the implementation of a (near) real-time FGO-NAV solution.

To fulfil these targets, the following phases were respected:

1. State-of-the-art review of FGO techniques and solutions in the PNT domain.
2. Design and implementation of a flexible FGO-NAV software unit.
3. Design and implementation of a testbed to benchmark the performance of the developed FGO-NAV software unit.
4. Functional tests and experimental campaign to assess the FGO-NAV performance in different operational scenarios.
5. Conclusions and recommendations for future activities.

State of the art – Review of FGO technics

– Link between FGO and GNSS

The traditional methods applied for GNSS positioning and navigation applications belong to the families of Weighted Least Squares (WLS) and Kalman Filters (KF).

These methods obey to assumptions and principles that are not always met in the real world. Thus, it was interesting to overcome these limitations with Factor Graph Optimization (FGO).

This highlights two main advantages of how FGO handles navigation constraints compared to traditional methods:

- FGO operates on a batch of historical data, allowing temporal correlations between the measurements. This provides robustness against non-Gaussian errors such as non-line-of-sight (NLOS) signals and multipath effects, which often exhibit temporal and spatial correlations.
- FGO performs multiple linearization iterations at each time step, improving the handling of nonlinearities that can amplify the effects of non-Gaussian errors.

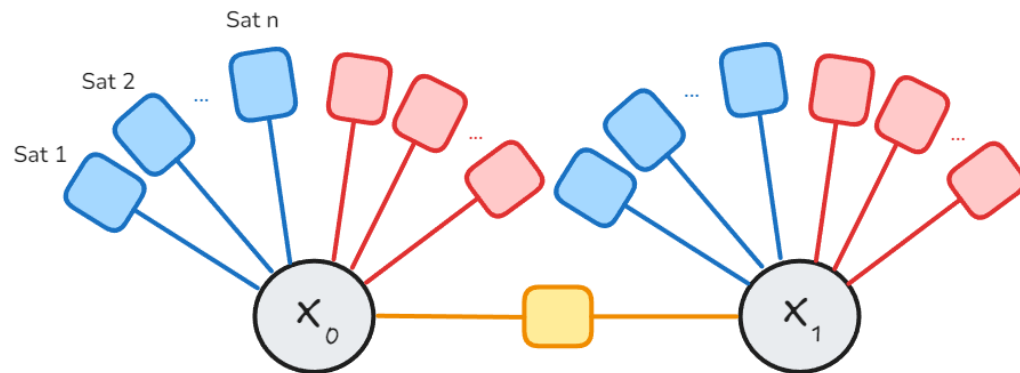
The benchmark of different implementations of FGO allowed us to identify an FGO method based on iSAM algorithm as more suitable choice for our GNSS context because iSAM algorithm is an immediate step toward real time PVT estimation using FGO.

Additionally, we suggest the numerical stability of the iSAM operation and its coherence with the task at hand (incremental consideration of new measurements at each timestep) makes it a suitable choice for the optimization problem.

State of the art – Review of FGO technics

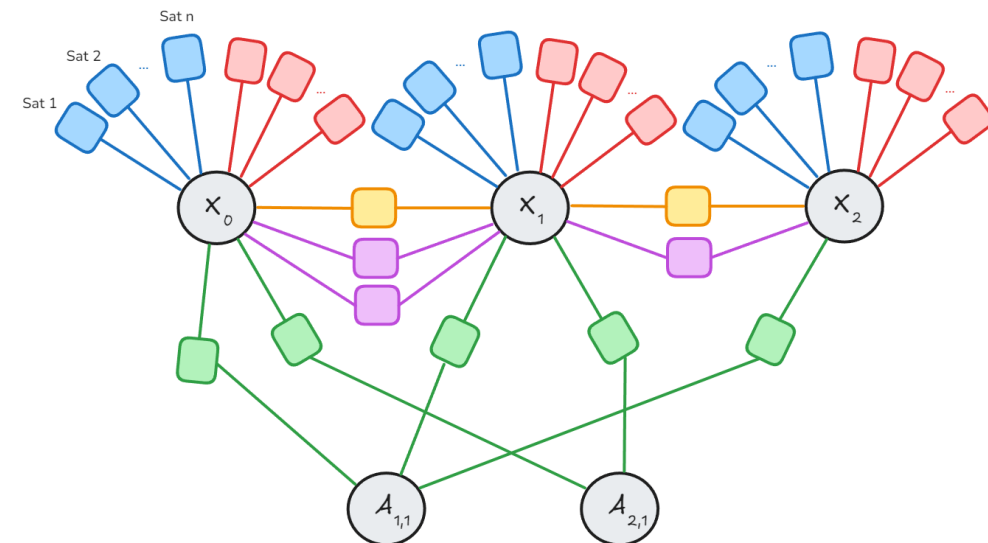
- Link between FGO and GNSS

Graph Model for FGO-SF and FGO-DF.



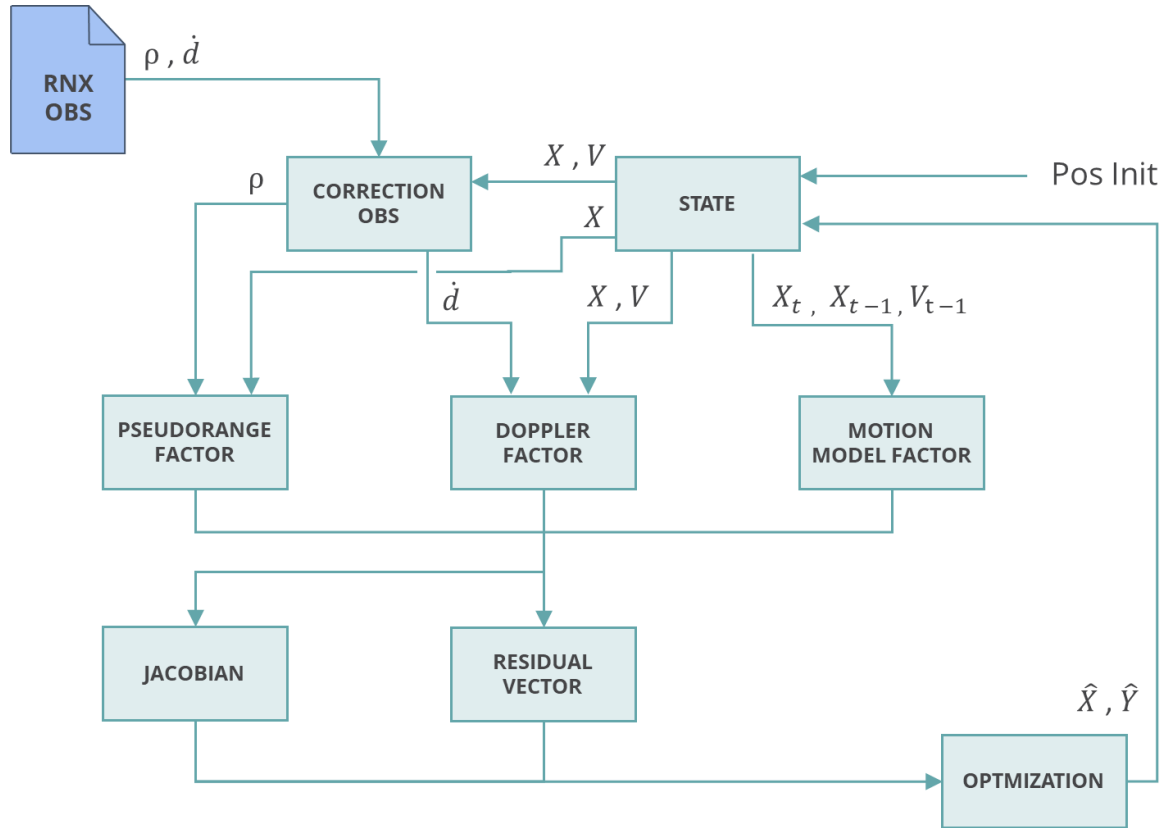
- Code Factor
- Doppler Shift Factor
- Dynamic Constraints

Graph Model for FGO-PPP



- Code Factor
- Doppler Shift Factor
- Dynamic Constraints
- Carrier-Phase Factor
- TDCP Factor

High-Level Design and implementation of a flexible FGO-NAV software unit

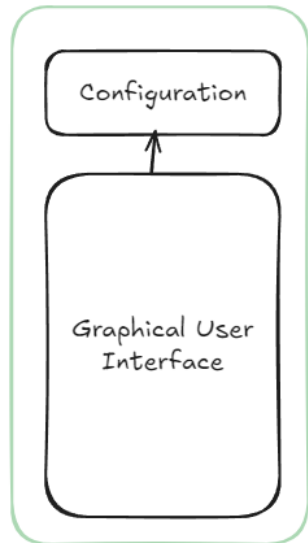


In order to have a **flexible software** unit the following choices were made:

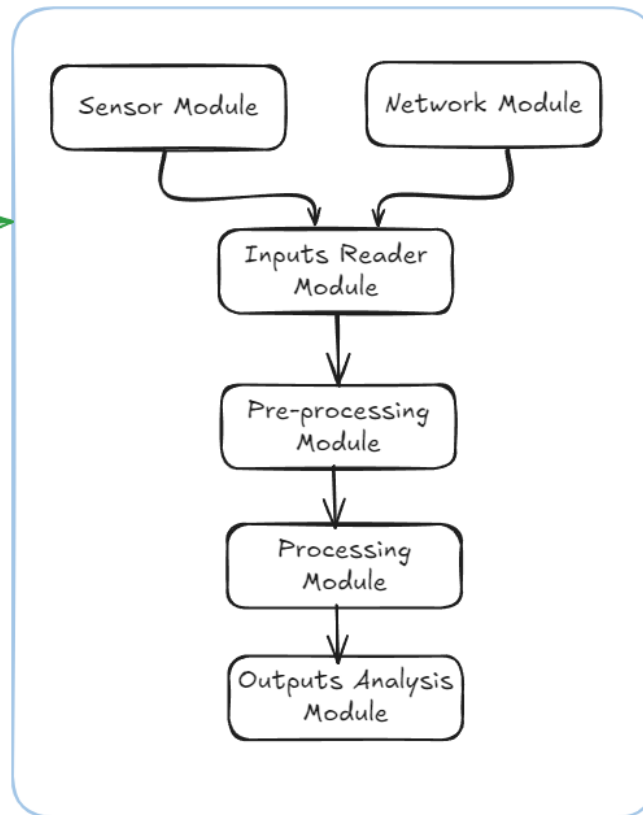
- Elementary operations are separated into **Functional Bricks (FUNB)**.
- A **Sequence of Operations** is configured to perform end-to-end computations. This sequence of Operations only involves FUNB.

High-Level Overview: Software Phases

CONFIGURATION



PROCESSING



SW Operations are separated into two phases:

- Configuration
- Processing

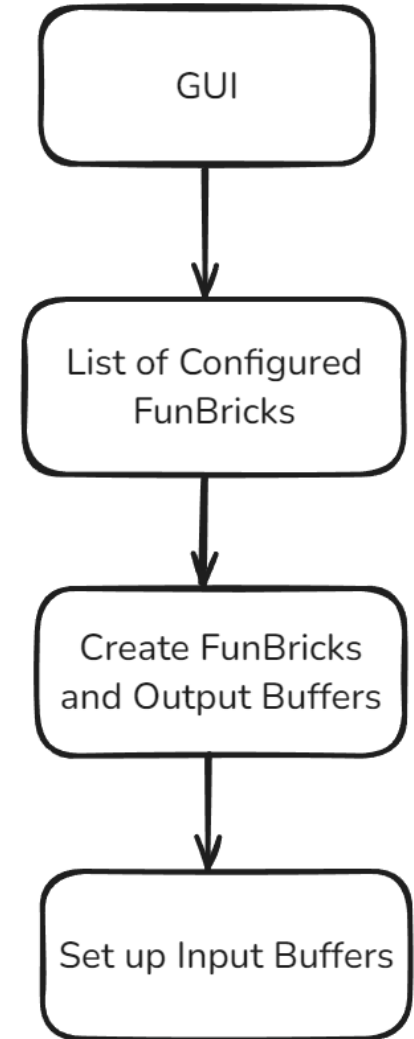
Configuration: Creation of the Harness

The objectives of the Configuration phase are:

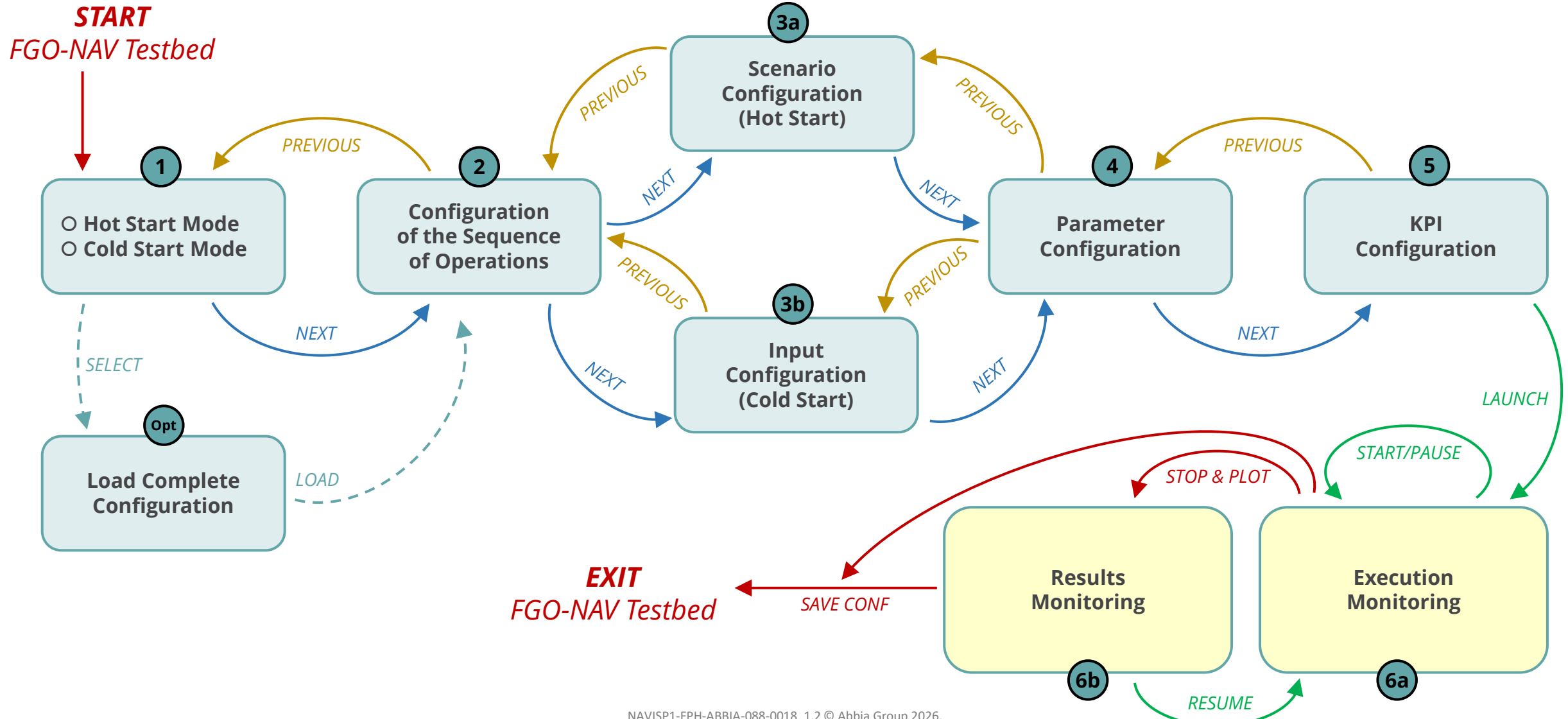
1. Creation of the Harness
2. Configuration of the Functional Bricks (Creation and Connection)

Operations are split into two phases:

1. Creation of Functional Bricks and Output Buffers
2. Configuration of Input Buffers



Concept of Operations driving the FGO-NAV SW architecture



FGO Algorithms implemented

The following FGO algorithms are implemented:

FGO-SF

Standalone GNSS Corrections. Ionosphere is corrected using the Klobuchar Model.

FGO-DF

Standalone GNSS Corrections except for Ionosphere which is corrected using Ionosphere-Free combinations of code measurements. If only Single frequency measurements are available, uses the Klobuchar Model.

FGO-INS-TC

Tightly coupled algorithm using GNSS and INS measurements. INS does not provide greater accuracy in nominal conditions as Doppler Shift based velocity estimation is very precise. But INS can help in very degraded conditions or during GNSS outages events.

FGO-PPP

Precise corrections of Code and Carrier Phase measurements. Carrier-phase is used to determine precise position estimates.

FGO-RTK

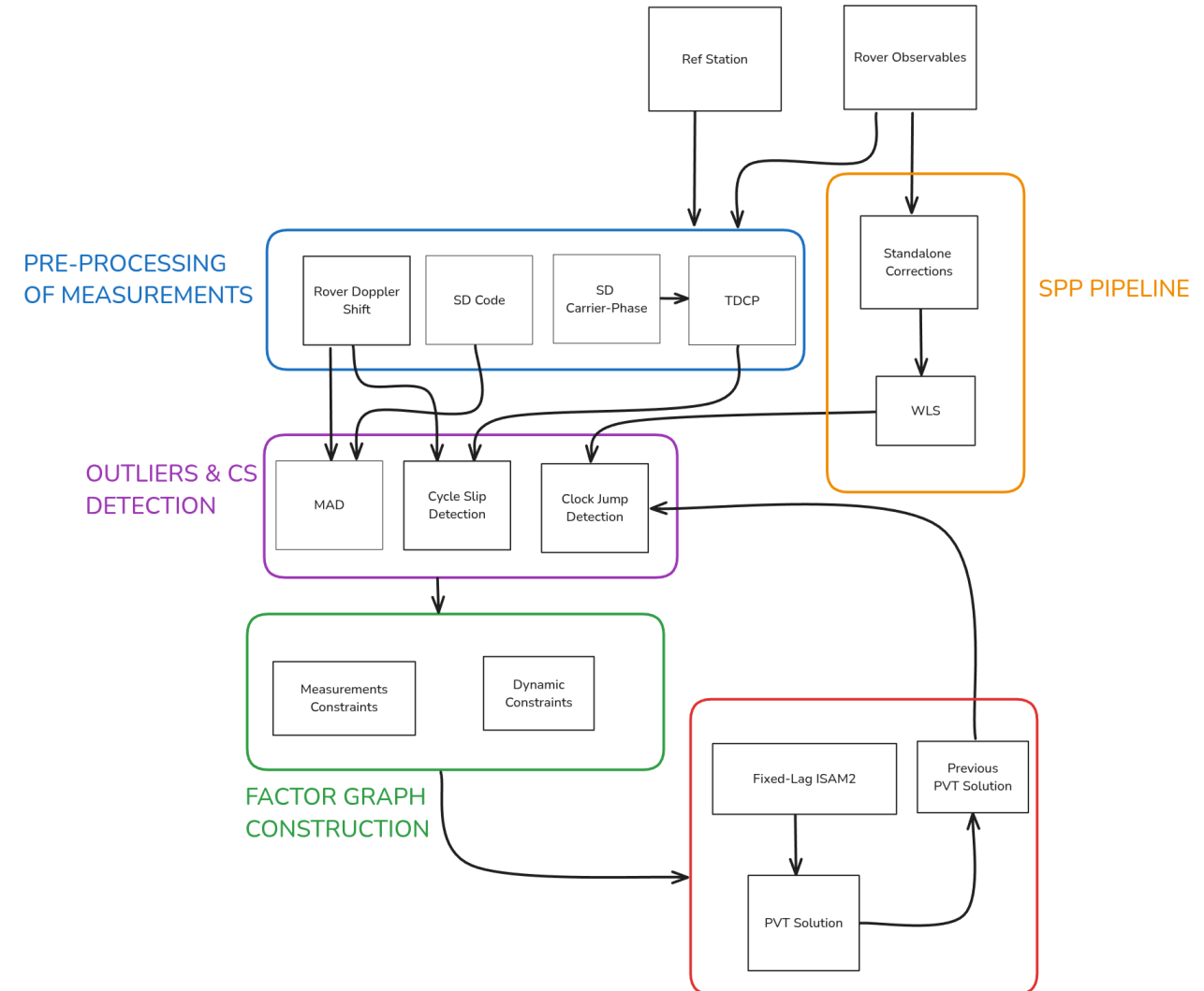
Uses measurements of a base station to correct measurements through Double Difference. Carrier-phase is used to determine precise position estimates.

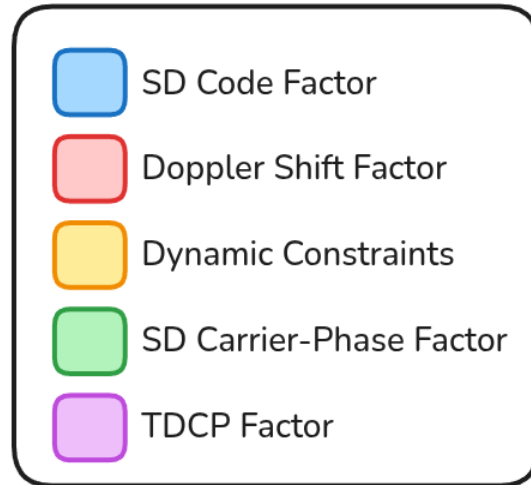
EKF variants of the PVT frameworks are also available.

SPP WLS is also implemented and compatible with FGO-SF and FGO-DF.

Robust FGO Algorithm

- We developed an additional robust PVT algorithm to provide precise PVT in harsh environments.
- This algorithm is derived from the most recent studies.
- We designed this algorithm for robust navigation in real-time.
- Uses measurements of a base station to perform Single Difference (SD) instead of Double Difference (DD).
- SD instead of DD:
 - No need to use a satellite as master satellite per constellation => more observations.
 - No introduction of noise through the master satellite if it is affected by multipath.
 - Less frequent cycle slips (CS), as DD requires to check CS in the master satellite carrier-phase measurements.

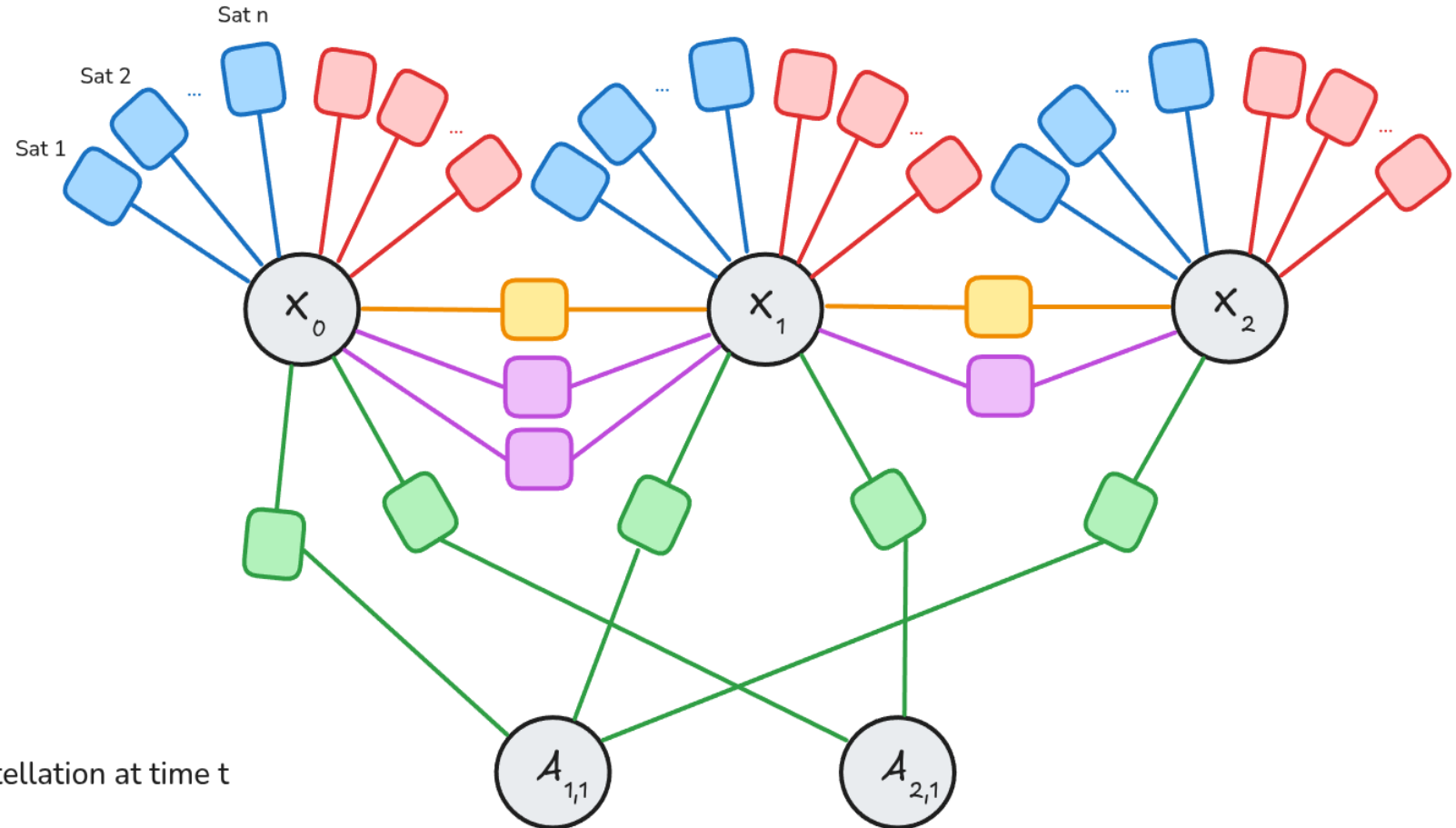




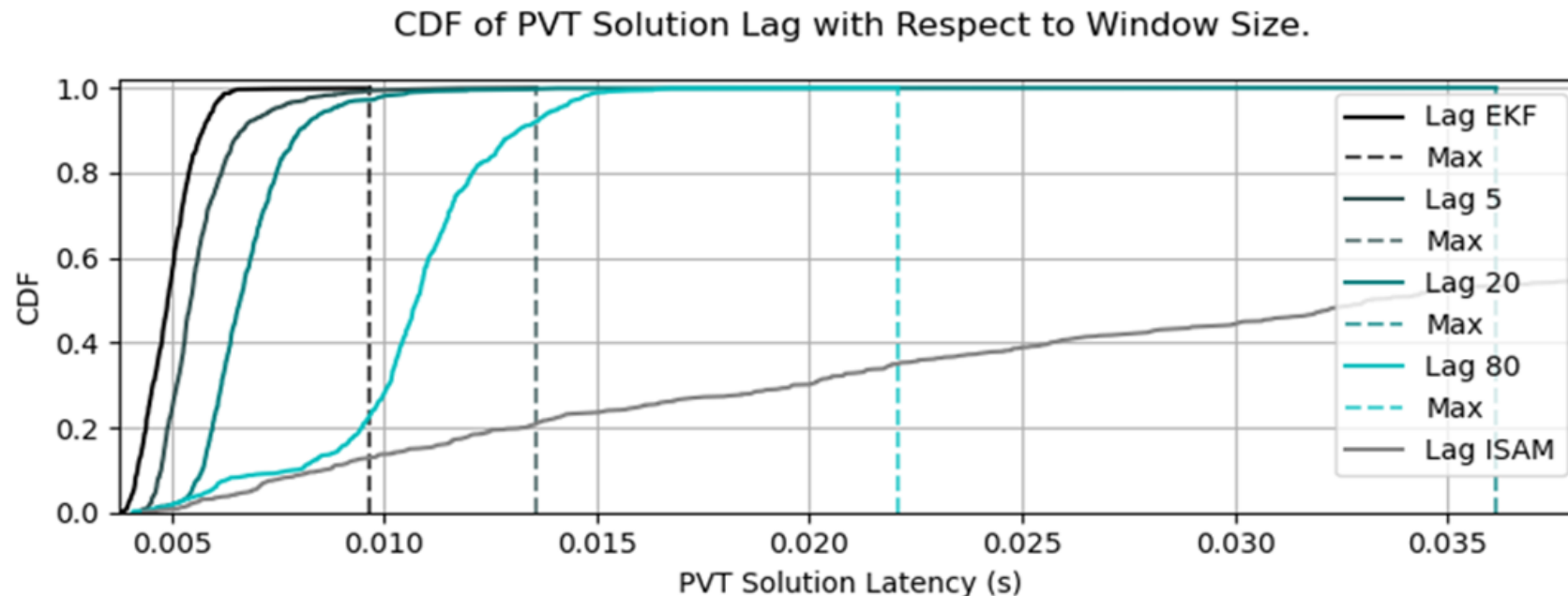
$$\chi_t = [X_t, V_t, C_t, D_t]$$

With:

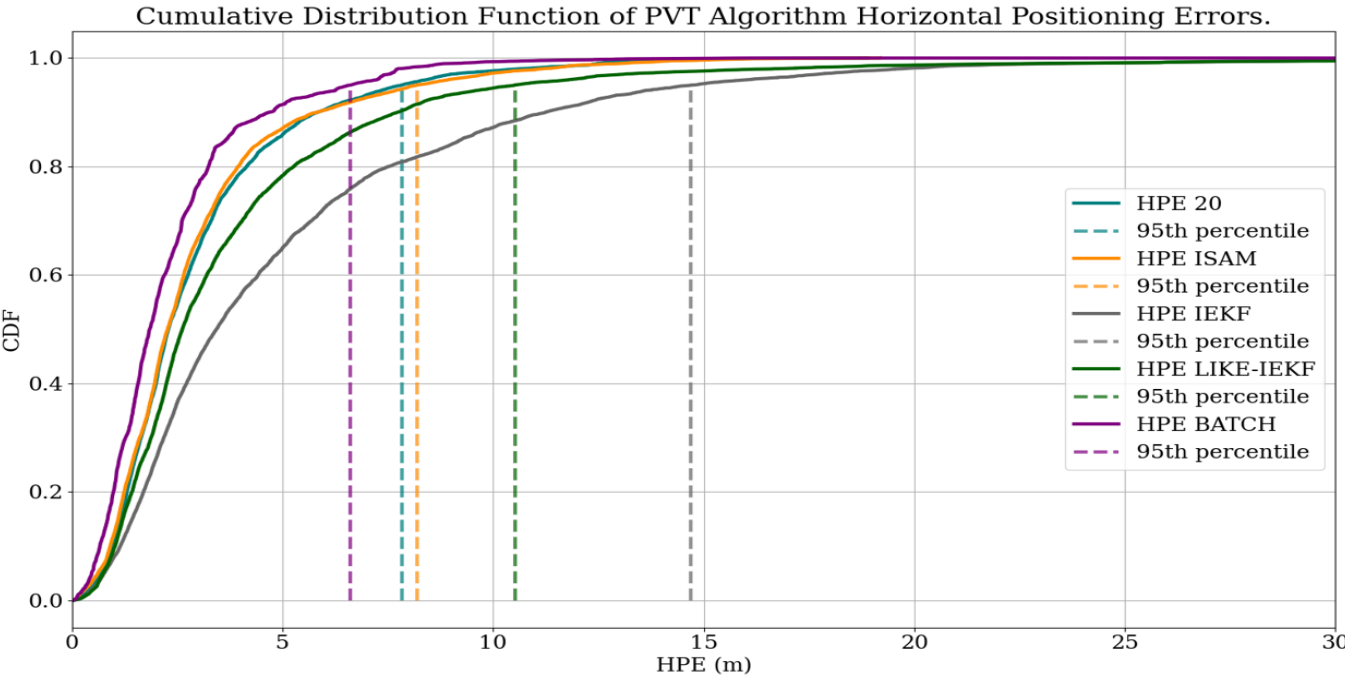
- X_t Position at time t
- V_t Velocity at time t
- C_t Clock Biases per Constellation at time t
- D_t Clock Drift at time t
- A_t Carrier-Phase Float Ambiguity at time t



- To achieve real-time capabilities. Our Factor Graph is solved through an Incremental solver called iSAM2.
- As the graph grows, latency of solutions becomes greater. For this reason, we used a fixed-lag smoother variant of iSAM2; which does not require relinearization of older measurements.
- **Our Algorithm is real-time capable.**



- We benchmarked different variants of our algorithm, changing optimisation methods.
- Notably to a robustified Incremental Extended Kalman Filter (IEKF) algorithm.
- FGO variants significantly outperforms the IEKF.



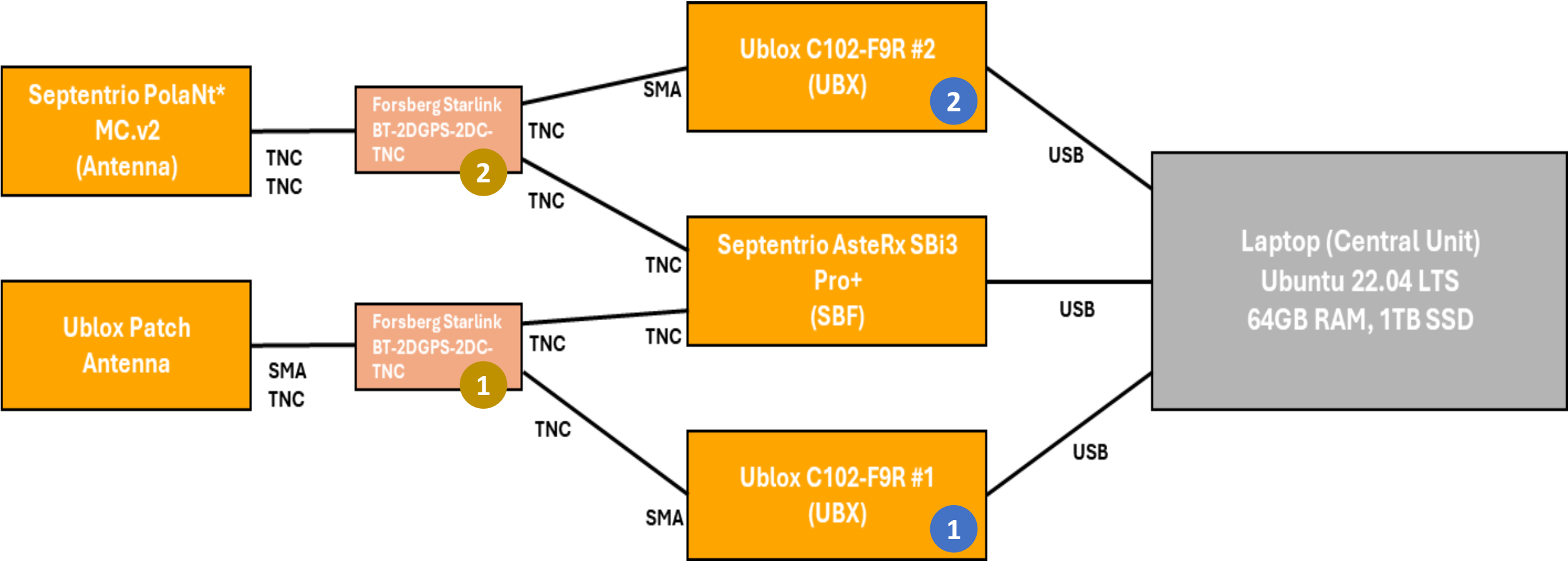
NAVISP1-FPH-ABBIA-088-0018_1.2 © Abbia Group 2026.

TABLE II. HORIZONTAL POSITIONING ERROR PER SCENARIO

Scenario	Method	HPE (m)			
		Mean	Std*	95 th Perc	Max
HK-Medium	FGO-IFLS	2.00	1.14	4.17	6.99
	FGO-IFLS-core	2.20	1.28	4.69	7.04
	IEKF	3.14	2.90	8.95	31.64
	iSAM2	1.90	1.11	4.07	7.01
	Batch FGO	1.19	0.66	2.20	6.44
HK-Deep	FGO-IFLS	2.23	1.57	5.31	8.29
	FGO-IFLS-core	3.07	2.27	7.32	13.46
	IEKF	4.60	3.75	13.17	18.46
	iSAM2	2.14	1.50	5.12	8.22
	Batch FGO	1.68	1.26	4.90	5.46
HK-Harsh	FGO-IFLS	4.01	2.86	9.96	15.97
	FGO-IFLS-core	6.01	6.70	18.52	46.77
	IEKF	6.38	5.66	19.54	29.56
	iSAM2	3.95	3.10	10.65	17.59
TK-Odaiba	Batch FGO	2.74	2.05	7.39	9.67
	FGO-IFLS	2.23	1.14	4.65	6.46
	FGO-IFLS-core	2.67	1.79	6.91	8.30
	IEKF	6.63	9.51	16.75	75.35
	iSAM2	2.23	1.19	4.87	7.09
TK-Shinjuju	Batch FGO	2.07	0.93	3.66	4.26
	FGO-IFLS	3.25	2.64	8.70	18.88
	FGO-IFLS-core	3.53	2.87	9.51	21.19
	IEKF	4.21	3.65	12.20	21.77
	iSAM2	3.13	2.62	8.79	18.74
	Batch FGO	3.03	2.38	8.12	15.98

*std = Standard Deviation

FGO-NAV testbed hardware interfaces



Testbed (operating view)

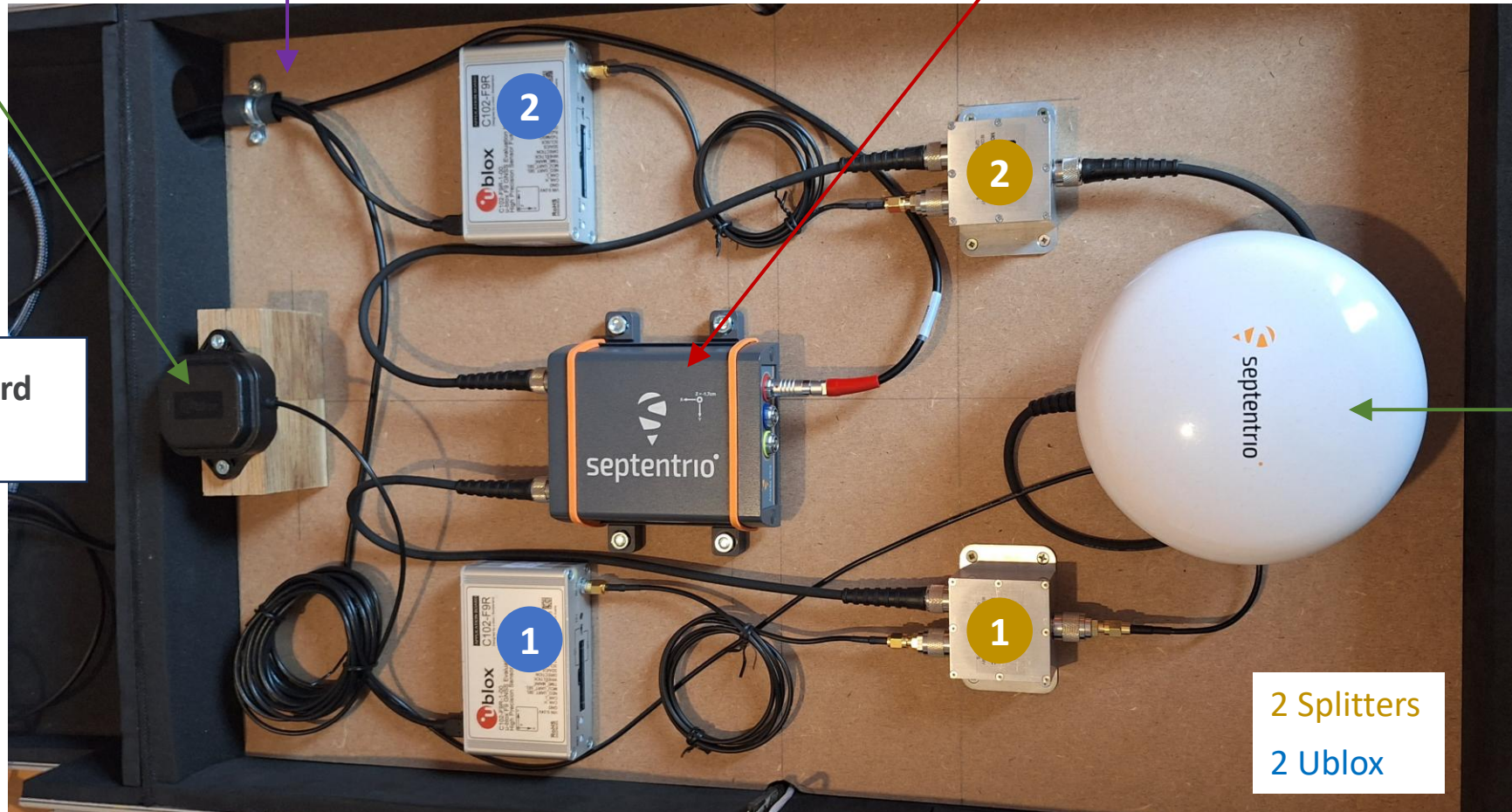
Patch antenna

Connection links with the laptop

Septentrio GNSS receiver (with integrated IMU)

The testbed is located and fixed on the roof of the vehicle used to carry out the data campaigns.

Vehicle Forward Movement

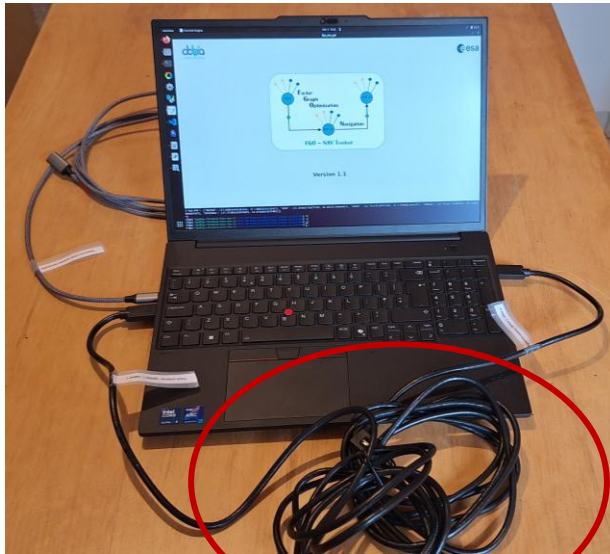


Septentrio antenna

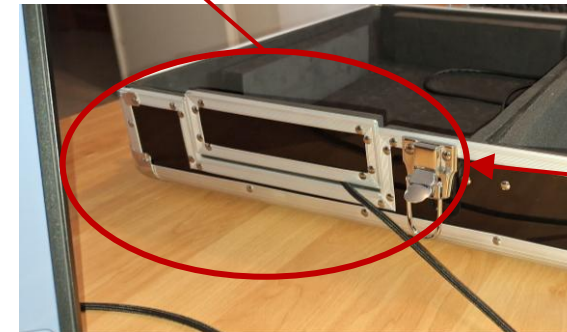
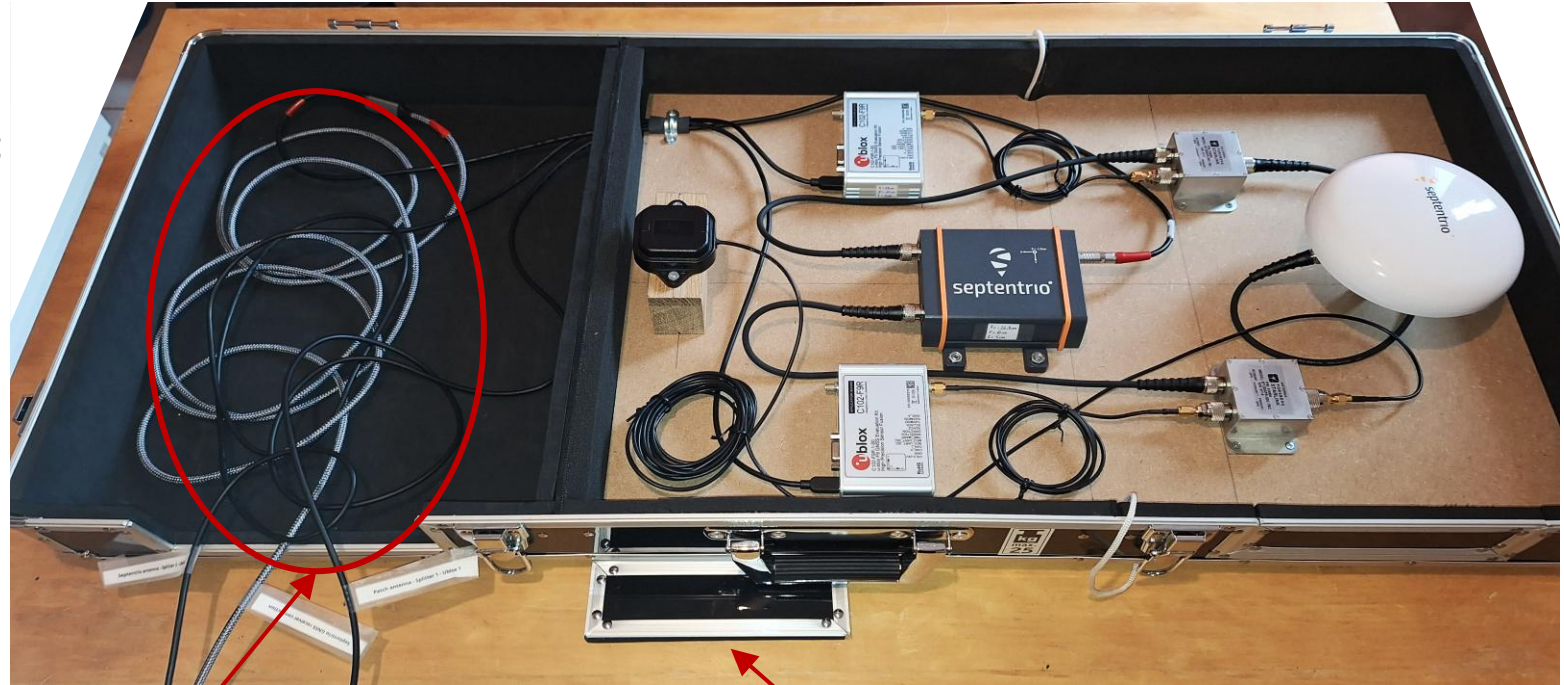
2 Splitters
2 Ublox

Testbed (operating view)

The laptop and its power supply are placed in the vehicle used to carry out the data campaigns.



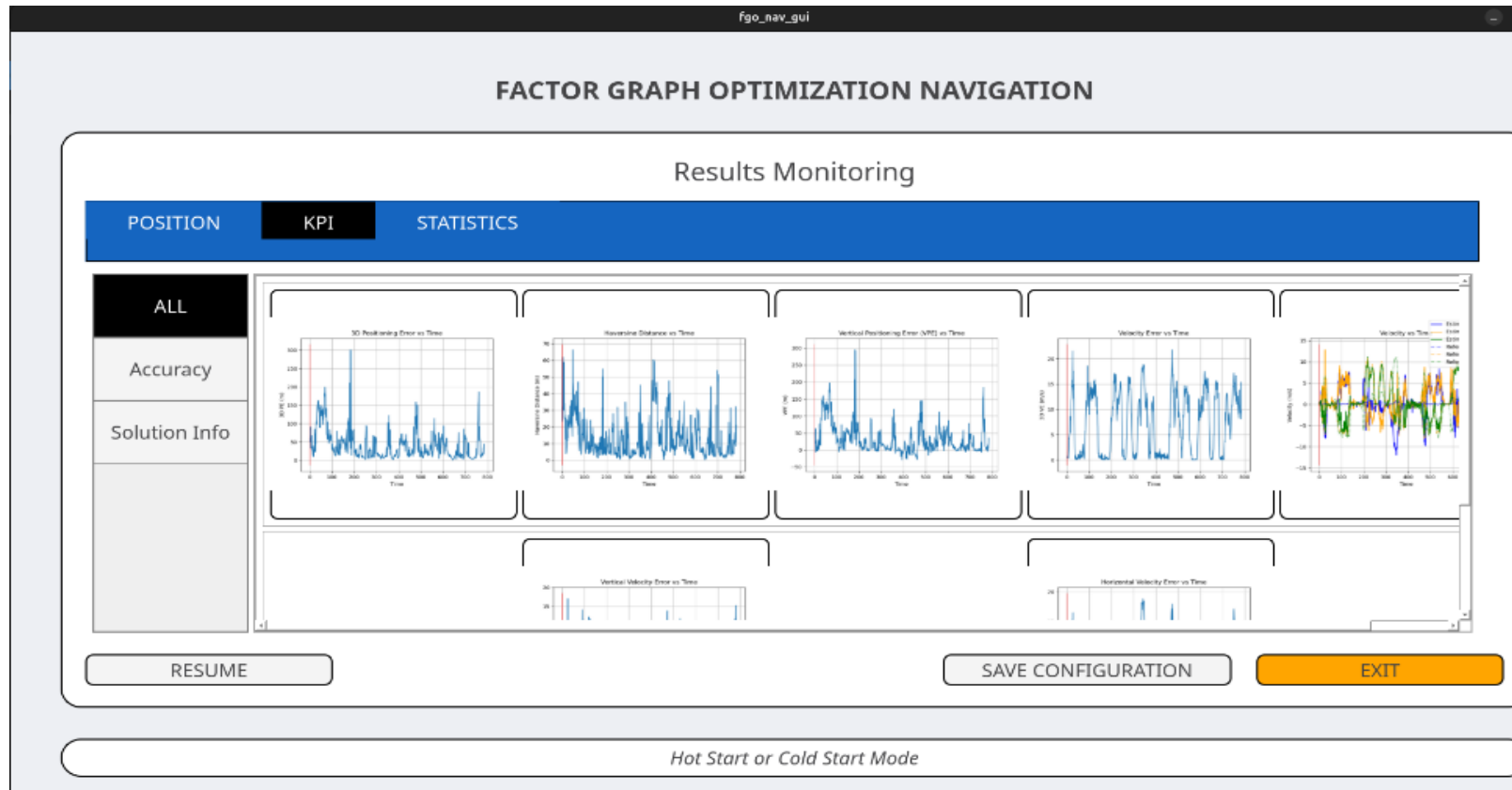
One 3-meter USB-C and two 3-meter USB cables are used to connect the testbed to power the components and enable the FGO-NAV software to operate.



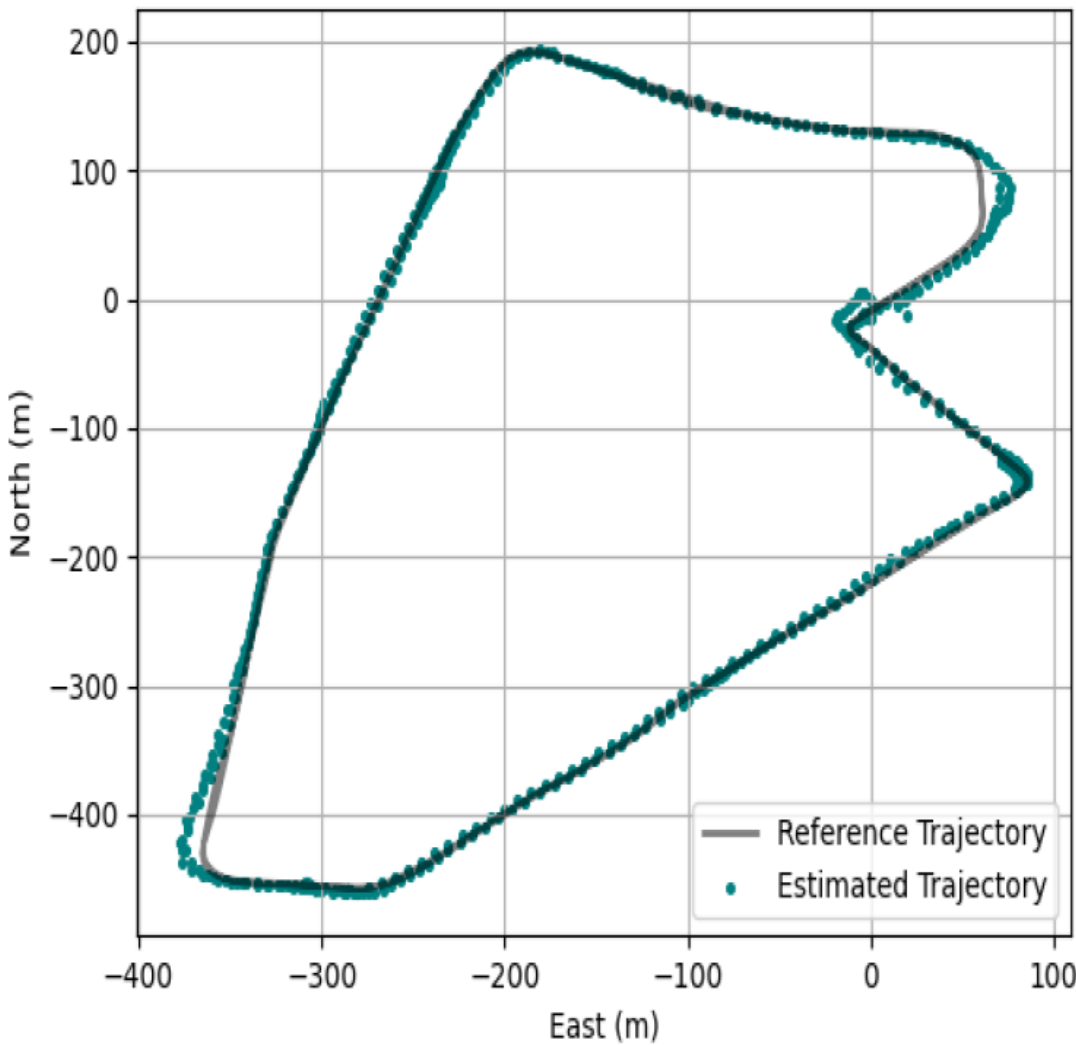
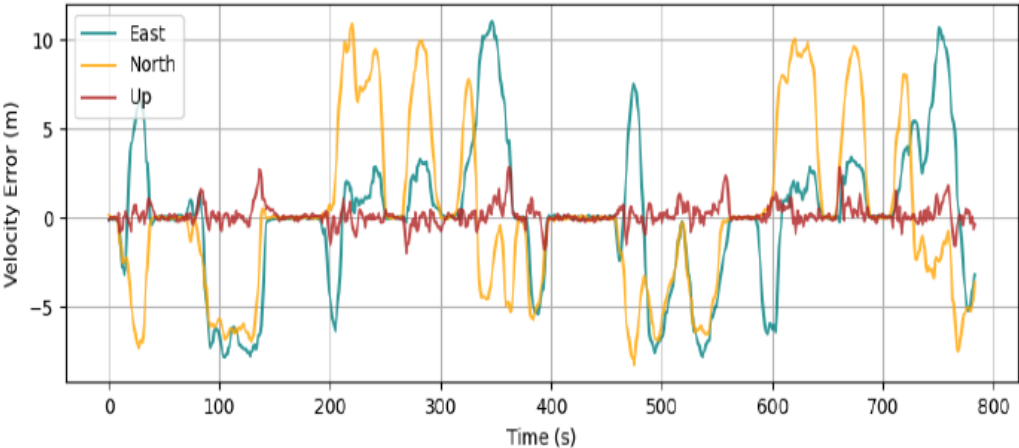
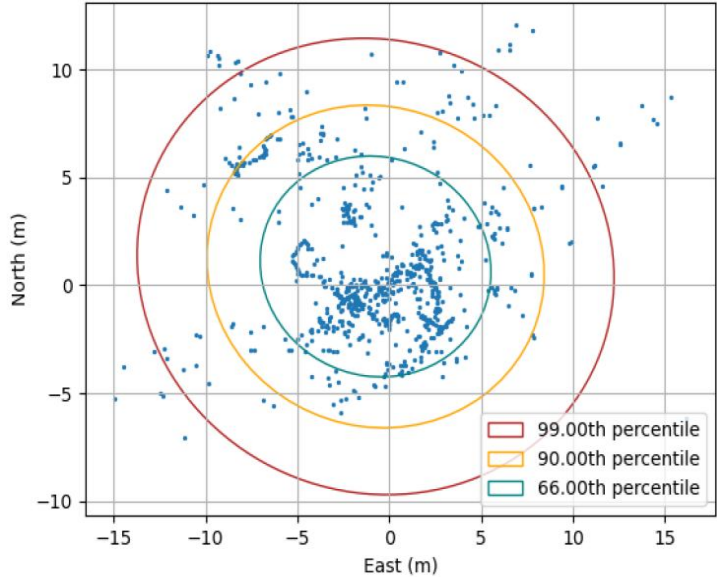
Removable front plate for cable routing with the vehicle.

GNSS Performances tools integrated in the testbed

- The testbed offers software tools allowing to benchmark the connected devices
- GNSS Receivers and IMU can be compared easily
- FGO methods can be benchmarked
- Software algorithms can be easily tested and benchmarked thanks to the flexible SW architecture designed

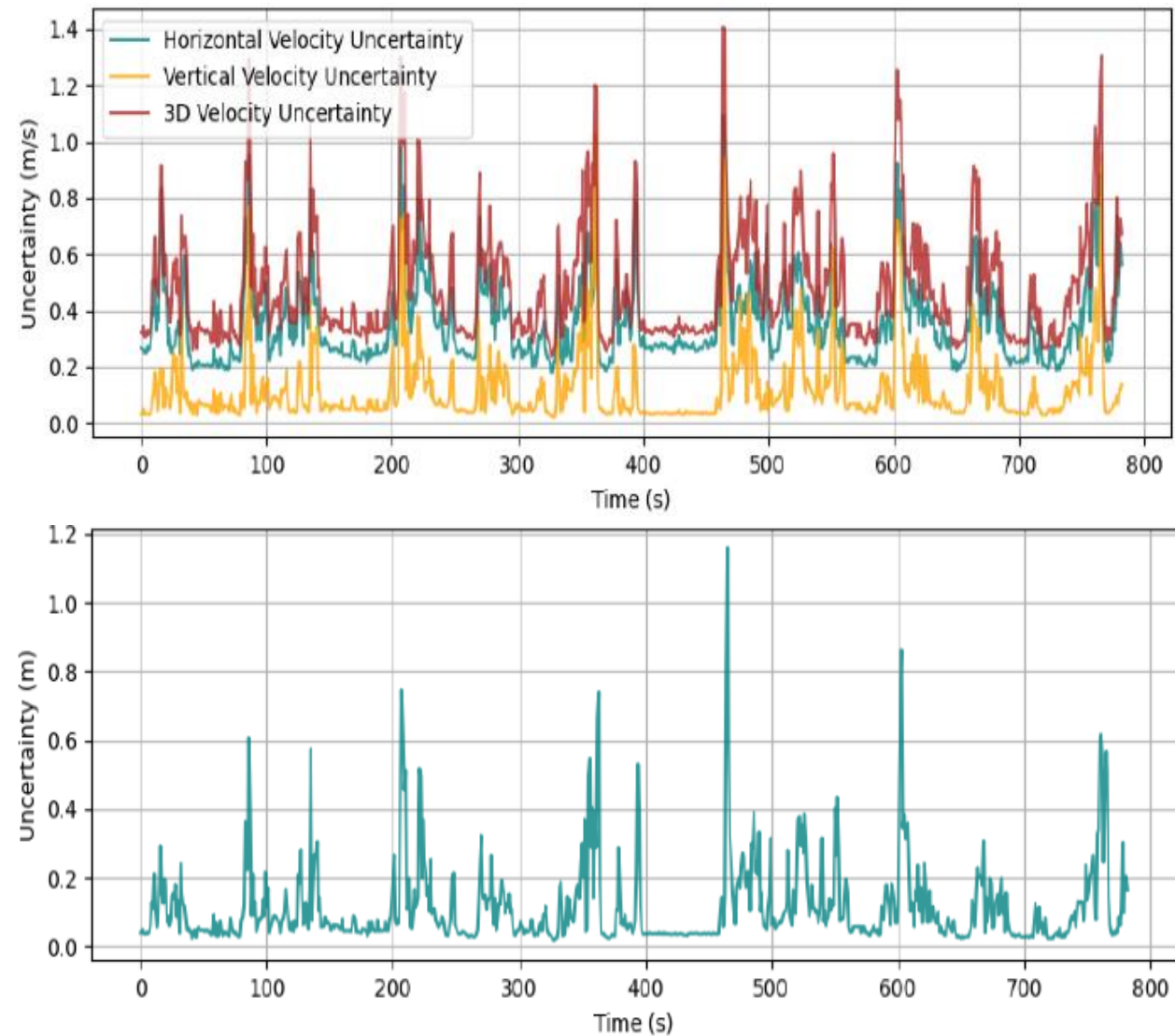
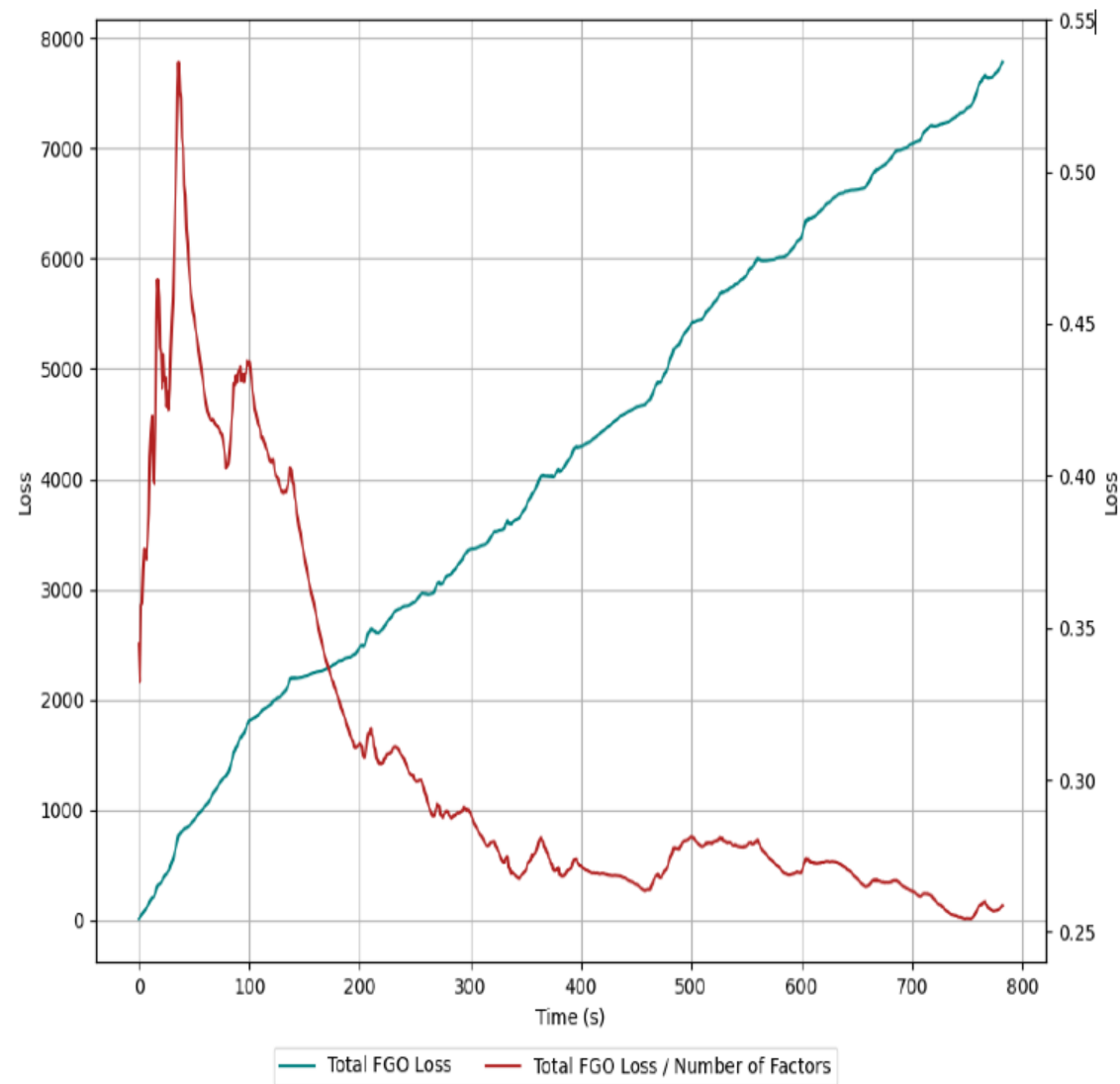


GNSS Performances tools integrated in the testbed



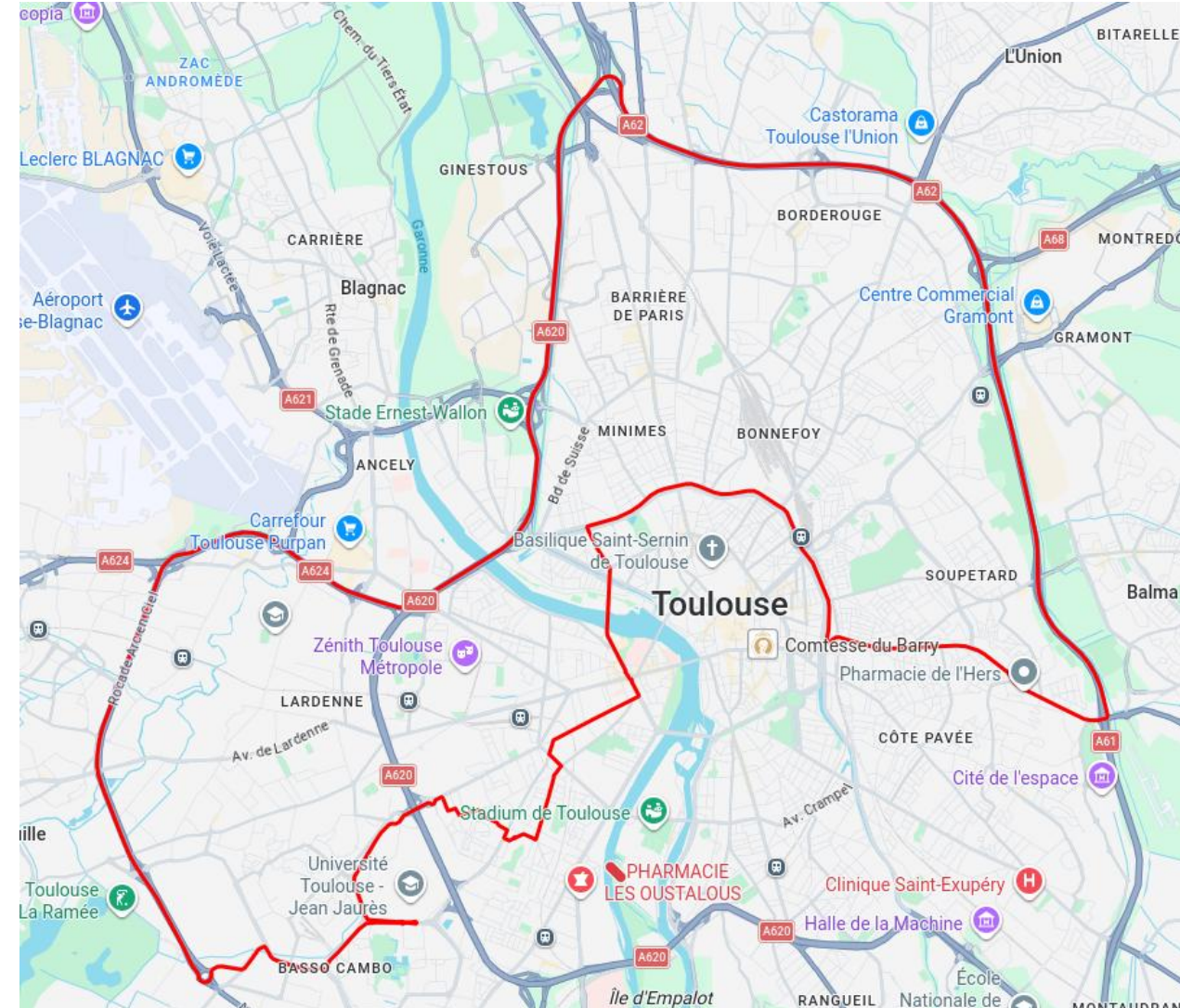
P1-FPH-ABBIA-088-0018_1.2 © Abbia Group 2026.

GNSS Performances tools integrated in the testbed



Results of the tests made in Toulouse city

- We developed a robust FGO-NAV algorithm to provide precise PVT in harsh environments
- This algorithm has been extensively tested in Toulouse urban area
- The circuit presented beside has been performed during different days, at different hours, with different traffic conditions
- This circuit has been selected because it offers different conditions of GNSS signal reception (open sky, urban canyons, short tunnels, mid-residential areas...)
- The testbed globally functioned as expected, but the low cost integrated GNSS receiver+IMU were not able to output the data at the sampling rate of 1 second
- A database of more than 24h of data captured with the FGO-NAV testbed is available

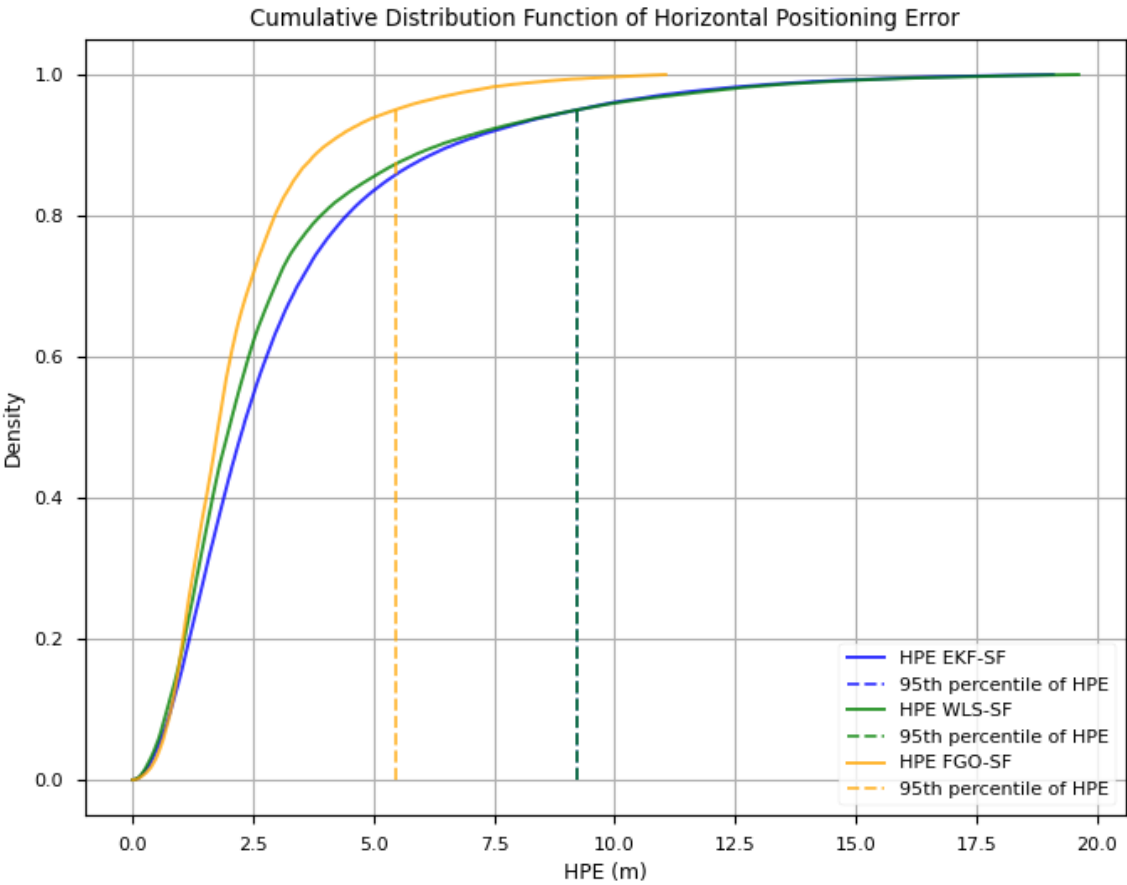


NAVISP1-FPH-ABBIA-088-0018_1.2 © Abbia Group 2026.

Results of the tests made in Toulouse city

- Our data campaign enabled a large-scale, meaningful comparison of:
 - Estimation frameworks (EKF, WLS and FGO), highlighting the advantage of FGO in dense urban conditions.
 - Positioning methods (INS fusion, RTK, etc.), highlighting the strengths and weaknesses of each approach.

Method	Mean	Std	95th percentile	99th percentile
WLS	3.22	5.15	10.01	19.64
EKF	3.44	4.79	9.91	19.09
FGO	2.37	3.69	5.89	11.06

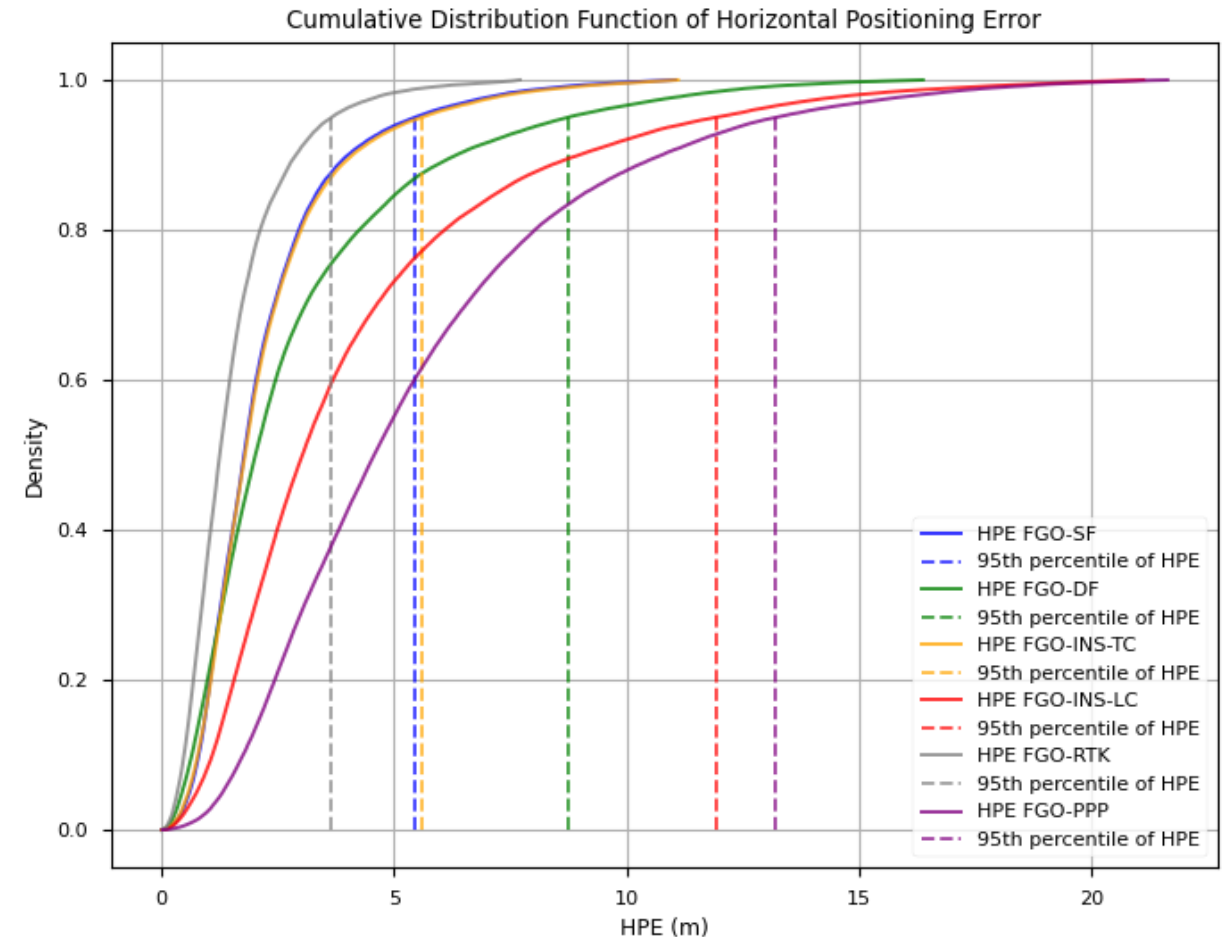


NOTE: Statistics and CDF are based on the entire data campaign (24h).

Results of the tests made in Toulouse city

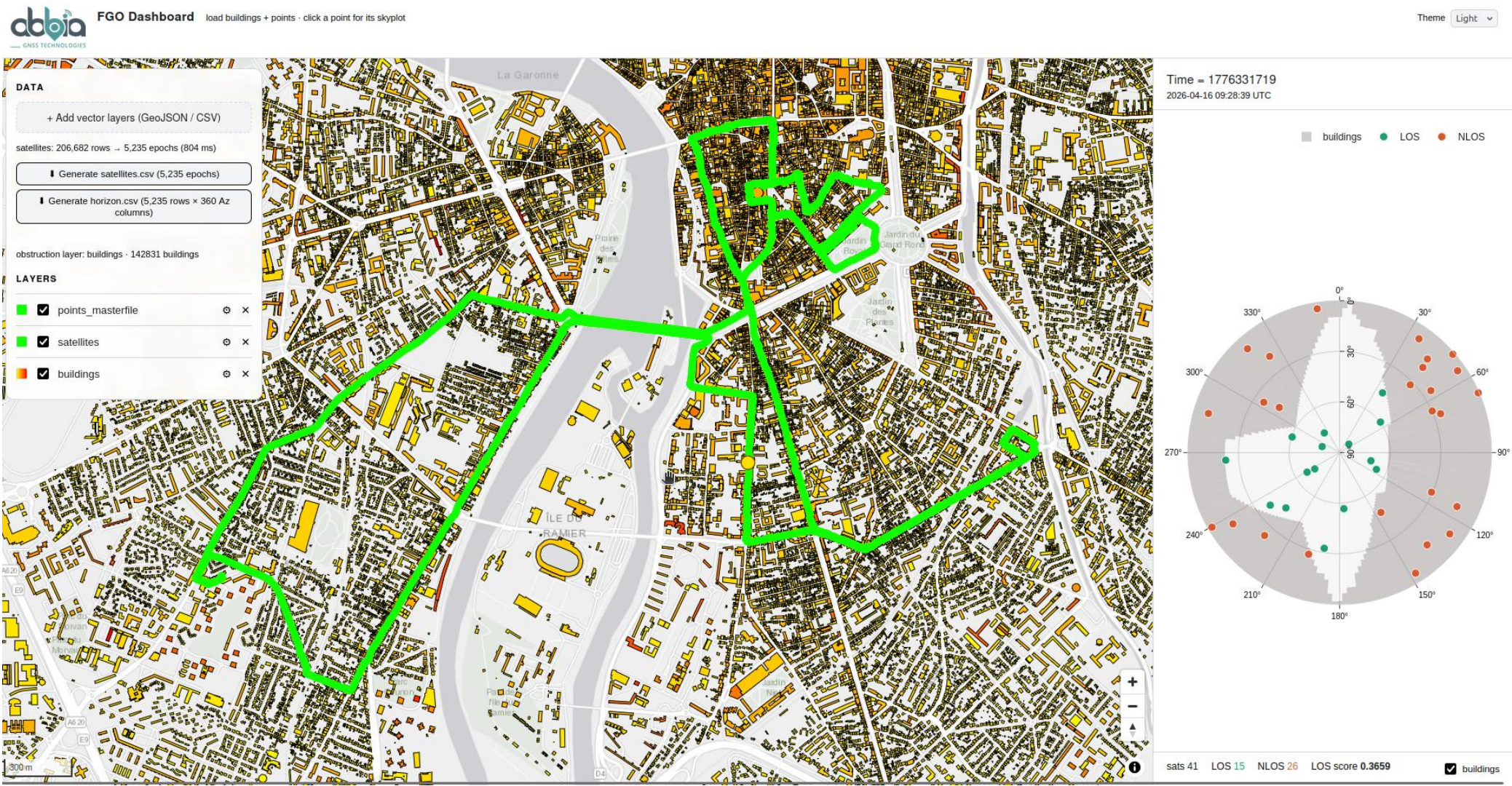
- Our data campaign enabled a large-scale, meaningful comparison of:
 - Estimation frameworks (EKF, WLS and FGO), highlighting the advantage of FGO in dense urban conditions.
 - Positioning methods (INS fusion, RTK, etc.), highlighting the strengths and weaknesses of each approach.

Method	Mean	Std	95th percentile	99th percentile
FGO-SF	2.37	3.69	5.89	11.06
FGO-DF	3.12	4.45	9.44	16.39
FGO-RTK	1.68	3.45	3.91	7.72
FGO-PPP	5,79	5,39	13,99	21,65
FGO-INS-LC	4.50	6.34	12.70	21.14
FGO-INS-TC	2.41	3.74	6.05	11.12



NOTE: Statistics and CDF are based on the entire data campaign (24h).

Results of the tests made in Toulouse city



Summary

- Abbia captured 24h of driving GNSS data in diverse environments.
- Diverse PVT algorithms based on EKF and FGO were evaluated.
- FGO has fundamental advantages over the EKF, especially for robust positioning / positioning in degraded measurements conditions; and for Sensor Fusion.

One of the modern challenges of GNSS is to **remain relevant in dense urban areas where multipath cannot be mitigated with traditional methods**; and where relative positioning sensors such as LiDAR and camera are largely preferred.

Measurement models and statistical tests eventually fail. Designing an algorithm that will operate accurately in diverse conditions requires to consider the case where the modelling assumptions will not hold. For this purpose, **the estimator can offer a last layer of defence.**

FGO might be the most promising foundation for modern positioning algorithms.

Conclusion and recommendations for future activities

Our work proposes a real-time, low-latency FGO-based PVT algorithm designed to remain resilient under harsh urban environments conditions. Low latency is achieved using the iSAM2 algorithm in a fixed-lag smoothing setting.

The algorithm was evaluated across several challenging urban scenarios, demonstrating convincing HPE performance, with a 95th percentile of 7.87 m across scenarios.

Benchmarking against a robust IEKF baseline confirmed the superiority of FGO, demonstrating a 46.5 % reduction in the 95th percentile HPE.

Although the EKF robustness can be improved through iterative updates, smoothing, or alternative modelling choices, such adaptations progressively narrow the conceptual and practical distinction with FGO. In contrast, FGO natively provides a unified framework in which smoothing, re-linearization and flexible modelling are naturally integrated.

These results highlight the strong potential of FGO for robust, real-time GNSS positioning in challenging urban environments.

Our recommendations for future technical activities are:

1. To improve the FGO-NAV algorithms by implementing an FGO-SBAS method and an FGO-HAS method.
2. To perform a deeper analysis of the 24 hours of the data capture in order to better understand the benefits offered by the FGO-NAV testbed in terms of multipath detection, interferences detection.
3. To improve the real-time GNSS positioning capability by porting the FGO-NAV software on a software embedded version and installing it on an electronic card designed for real time embedded software.
4. To embed the FGO-NAV improved software on a low consumption GNSS receiver chipset.

In parallel to the technical activities recommended, it is also important to explore the commercial interest of having the real-time FGO-NAV GNSS positioning for the urban navigation of automated mobiles drones.



contact@abbia.fr
+33 (0) 534 612 544

*27, rue Jules Amilhau
31100 Toulouse – France*

www.abbia.fr/abbia-gnss/