Outdoor Ground Truth Estimation via GNSS Carrier Phase and Factor Graph Optimization

Hakan Uyanik[1], Steffen Schön[2], Daniel Medina[1]

***Abstract*— Assessing the quality of a positioning solution is essential for validating high precision GNSS algorithms. Generating a reliable reference trajectory is often challenging because it usually requires expensive sensor units, careful calibration, and independent validation. This paper evaluates several estimators for precise GNSS based trajectory generation using an undifferenced multi frequency PPP float observation model. The compared estimators are an error state Kalman filter (ESKF), a Rauch Tung Striebel (RTS) smoother, sliding window factor graph optimization (FGO), and full batch FGO. The objective is to identify which estimator provides the highest post processed precision from the available GNSS data while also highlighting the practical drawbacks of estimators, especially in terms of computational cost and accuracy. All implementations use the same data handling, observation model, stochastic settings, process model, and float ambiguity handling. The benchmark uses an open sky kinematic GNSS data set of 6000 epochs sampled at 2 Hz. A two epoch sliding graph behaves close to the forward ESKF, a 30 epoch graph provides measurable improvement, and the full batch graph gives the strongest post processed trajectory. The study clarifies the accuracy and cost tradeoffs of FGO and shows why full batch optimization is a suitable reference quality post processing estimator for trajectory generation with GNSS.**

I. INTRODUCTION

Reliable reference trajectories are essential for developing and validating navigation solutions, perception systems, and autonomous platforms. A positioning error that may be acceptable for coarse localization can become relevant when the trajectory is used for estimator benchmarking, sensor fusion tuning, mapping, or safety critical replay analysis. The practical question is therefore focuses on how to provide a low latency navigation output and also how to generate the most consistent post processed trajectory from the available GNSS measurements. This requirement naturally points to estimators that use individual epoch measurements and the temporal relationships between observations, nuisance states, and carrier phase ambiguities.

Kalman filtering has been the conventional estimator family for GNSS and integrated navigation because it provides a recursive and efficient solution for dynamic state estimation [1], [2]. RTS smoothing extends the filter by applying a backward pass to the stored forward solution [3]. It is a strong offline baseline because future data influence earlier states, but the smoother still inherits the state transition, covariance, and linearization history of the forward filter. Factor graph optimization provides a broader smoothing formulation. Taylor and Gross present factor graphs as a navigation estimation framework that generalizes Kalman filtering, constructs an associated optimization problem, and exploits sparse linear algebra for efficient computation [7]. Dellaert and Kaess describe the factor graph representation for robot perception, while iSAM2 introduced incremental smoothing through Bayes tree updates [4], [5]. Beyond GNSS, FGO has become a popular solution for navigation, localization, and mapping problems, especially in robotics applications where nonlinear constraints and heterogeneous sensor factors must be optimized jointly.

In the GNSS community, the work of Wen, Hsu and coauthors has been influential in bringing FGO based GNSS estimation to wider attention. Wen and Hsu formulated robust GNSS positioning and RTK using factor graph optimization, showing its potential under urban conditions [10]. Wen, Pfeifer, Bai and Hsu compared FGO against EKF for GNSS/INS integration and showed how the graph formulation can exploit time correlated information in integrated navigation [11]. Bai, Wen and Hsu further studied time correlated window carrier phase aided GNSS positioning using FGO for urban positioning [12]. These works are important because they connect the general graph optimization machinery known from robotics to carrier phase GNSS, where ambiguity handling, cycle slip detection, and measurement quality control dominate the achievable accuracy.

The practical value of FGO for GNSS trajectory estimation under challenging measurement environments are also studied. Suzuki demonstrated graph based optimization for smartphone GNSS using corrected pseudorange, accumulated delta range, Doppler, TDCP, IMU, and related constraint factors [22], [23]. Zhang et al. proposed continuous time GNSS and multisensor FGO, fusing asynchronous GNSS, speed, IMU, and lidar odometry measurements through a Gaussian process trajectory representation [24]. These studies are complementary to the present work, which keeps the setup GNSS only and PPP float to evaluate the contribution of carrier phase based FGO before adding additional sensors or ambiguity fixing.

Recent high precision GNSS FGO studies show that the largest benefits often appear when graph memory is combined with carrier phase continuity, ambiguity resolution, robust factor design, or correction services. Li et al. developed an FGO based PPP RTK framework in which raw pseudorange, carrier phase, precise atmospheric corrections, and time differenced carrier phase measurements serve as factors. They report that continuously tracked

[1]Institute of Communications and Navigation, German Aerospace Center (DLR), Neustrelitz, Germany. Email: ahmet.uyanik@dlr.de,
[2]Bundesamt für Kartographie und Geodäsie (BKG), Frankfurt, Germany,
[2]Leibniz Universität Hannover, Germany

phase ambiguities are modeled as time invariant states and propagated by marginalization, and that a second optimization using ambiguity resolved solutions and TDCP improves reliability [13]. Wang et al. studied multi GNSS RTK in urban canyons using sliding window FGO and showed that ambiguity propagation can fully use continuously tracked carrier phase information. In deep urban conditions, their FGO RTK variant with ambiguity propagation improved 3D accuracy by 69.6 percent compared with EKF RTK and reduced the dependence on window size [14]. Xiao et al. formulated PPP ambiguity resolution directly in a graph by constructing pseudorange, carrier phase, and ambiguity resolution factors from validated constraints [15]. Gao et al. also investigated robust GNSS RTK with ambiguity resolution in FGO and used a second optimization stage based on fixed integer ambiguities to improve fix performance [16]. The comparative analysis of carrier phase based precision navigation emphasizes the same design space: FGO is flexible for combining motion constraints, ambiguity modeling, robust optimization, and multiple sensor modalities, but the tradeoff between accuracy, robustness, and computational cost must be selected for the application [17].

Robustness against faults and contaminated measurements is indispensable for GNSS navigation in urban and other degraded scenarios. Medina et al. studied robust statistics for GNSS single point positioning and discussed how robust regression can mitigate contaminated GNSS observables [18]. Bellés et al. applied robust statistics to GNSS joint position and attitude estimation in harsh environments [19], and Bellés and Medina later proposed robust PPP AR using M estimators for multi fault scenarios [20]. These works support the broader interpretation that an FGO based reference trajectory framework becomes more valuable when open sky assumptions fail, because contaminated measurements, non line of sight reception, cycle slips, and intermittent visibility can be handled through robust factors, validated ambiguity constraints, and additional sensor or correction factors.

In this work, we propose an FGO based solution for reference quality GNSS trajectory generation and assess its performance against conventional recursive estimators, namely the ESKF and the RTS smoother. The goal is to show the accuracy gain of full batch optimization and to evaluate different estimator choices and expose the drawbacks of FGO, including increased runtime and the approximation introduced by sliding window marginalization. The experiment is first conducted on an open sky kinematic data set. This setting does not fully exploit the strongest advantages of FGO, since severe multipath, blockage, and intermittent carrier phase tracking are limited. However, it provides a necessary sanity check: with the same PPP float model, a very short graph should remain close to a well tuned ESKF. Any systematic improvement from longer windows can then be attributed mainly to the ability of the graph to distribute residual biases, ambiguity coupling, and temporal correlations over a longer trajectory segment.

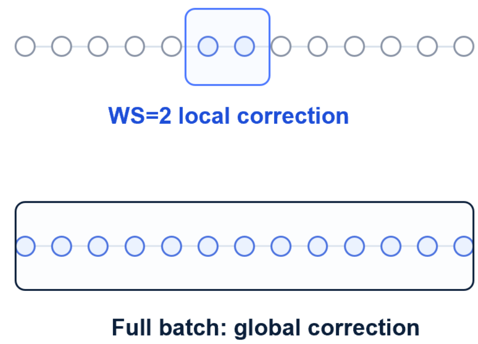

Fig. 1. Window size and full batch representation.

The contribution of this study is a controlled estimator comparison for high precision GNSS trajectory generation. Unlike many GNSS FGO studies that combine graph optimization with integer ambiguity resolution, PPP RTK corrections, robust switching, Doppler or time differenced carrier phase factors, or additional inertial and vision sensors, this work keeps the single antenna undifferenced multi frequency PPP float setup fixed. The observation model, stochastic settings, process model, preprocessing, and float ambiguity handling are common to all variants. The results identify three regimes. A two epoch graph behaves close to the forward ESKF, a 30 epoch graph gives measurable improvement, and full batch FGO gives the strongest post processed trajectory among the tested variants.

Fig. 1 shows the representation of measurement window used for the evaluation.

II. METHODOLOGY

A. PPP Float Model

All estimators use the same undifferenced multi frequency PPP float observation model. Let $\mathcal{F}$ denote the carrier frequencies and $\mathcal{S}_k$ the satellites used at epoch k. For satellite $s \in \mathcal{S}_k$ and frequency $f \in \mathcal{F}$, the code and carrier phase observations are

$$\begin{aligned} P_{k,f}^s &= \rho_k^s + b_{r,k}^{g(s)} - b_k^s + m_{T,k}^s Z_{w,k} + \gamma_f I_k^s + \varepsilon_{P,k,f}^s, \\ L_{k,f}^s &= \rho_k^s + b_{r,k}^{g(s)} - b_k^s + m_{T,k}^s Z_{w,k} - \gamma_f I_k^s + \lambda_f N_{k,f}^s + \varepsilon_{L,k,f}^s. \end{aligned} \tag{1}$$

Here ρ_k^s is the geometric range, $b_{r,k}^{g(s)}$ is the receiver clock bias for constellation $g(s)$, b_k^s is the satellite clock correction, $Z_{w,k}$ is the zenith wet delay, $m_{T,k}^s$ is the mapping coefficient, I_k^s is the slant ionospheric delay on the reference frequency, λ_f is the wavelength, and $N_{k,f}^s$ is the float carrier phase ambiguity. The sign difference of the ionospheric term between code and carrier phase follows the standard dispersive GNSS observation model [8], [9].

The epoch state is

$$\begin{aligned} \boldsymbol{x}_k = \big[&\boldsymbol{p}_k^T,\ \boldsymbol{v}_k^T,\ b_{G,k},\ b_{E,k},\ \dot{b}_{G,k},\ \dot{b}_{E,k},\ Z_{w,k}, \\ &I_k^1, \ldots, I_k^{n_k},\ N_{k,1}^1, \ldots, N_{k,f}^{n_k}\big]^T. \end{aligned} \tag{2}$$

The same state definition, satellite corrections, stochastic settings, and cycle slip handling are used by the ESKF, RTS, and FGO runs. No integer ambiguity fixing is used in the reported results.

B. Filter and Smoother Baselines

The ESKF is used as the recursive baseline. It propagates $\hat{\boldsymbol{x}}_k$ and $\boldsymbol{P}_k$ forward, linearizes the stacked code and carrier residual $\boldsymbol{y}_k = \boldsymbol{z}_k - \boldsymbol{h}(\hat{\boldsymbol{x}}_k^-)$ as $\boldsymbol{y}_k \simeq \boldsymbol{H}_k \delta \boldsymbol{x}_k + \boldsymbol{\eta}_k$, and applies the standard Kalman update. In this GNSS only setup, the correction is additive for position, velocity, clock, atmosphere, ionosphere, and float ambiguity states. The RTS smoother is the offline baseline obtained by applying a backward pass to the stored ESKF means, covariances, and transition matrices. Therefore, RTS uses future information, but it still inherits the covariance and linearization history of the forward filter.

C. Graph Formulation

FGO estimates the active trajectory and nuisance variables by minimizing a weighted residual objective:

$$\begin{aligned} \mathcal{X}^\star = \arg\min_{\mathcal{X}} \Big[&\|\boldsymbol{r}_0\|^2_{\boldsymbol{\Sigma}_0^{-1}} + \sum_k \|\boldsymbol{r}_{Q,k}\|^2_{\boldsymbol{Q}_k^{-1}} \\ &+ \sum_k \rho_\kappa \Big(\|\boldsymbol{r}_{P,L,k}(\mathcal{X})\|^2_{\boldsymbol{R}_k^{-1}} \Big) \Big]. \end{aligned} \tag{3}$$

The first term is the prior, the second term is temporal process consistency, and the third term contains the stacked code and carrier phase factors. The robust loss ρ_κ and covariance settings are common to the compared estimators. At iteration ℓ, the local least squares problem is

$$\Delta \mathcal{X}^\ell = \arg\min_{\Delta \mathcal{X}} \left\| \boldsymbol{r}^\ell + \boldsymbol{J}^\ell \Delta \mathcal{X} \right\|^2_{\boldsymbol{\Omega}}, \tag{4}$$

with normal equations

$$\boldsymbol{H}^\ell \Delta \mathcal{X}^\ell = -\boldsymbol{g}^\ell, \qquad \boldsymbol{H}^\ell = (\boldsymbol{J}^\ell)^T \boldsymbol{\Omega} \boldsymbol{J}^\ell, \quad \boldsymbol{g}^\ell = (\boldsymbol{J}^\ell)^T \boldsymbol{\Omega} \boldsymbol{r}^\ell. \tag{5}$$

This direct relinearization of connected code, carrier, process, ionosphere, and ambiguity residuals is the main reason why full batch FGO can differ from RTS smoothing: corrections are not confined to the filter history, and all active residuals can be adjusted jointly [4], [6], [7].

For a sliding window of length W, the active set at epoch k is

$$\mathcal{X}_k^W = \{\boldsymbol{x}_j : j \in [k - W + 1, k]\}. \tag{6}$$

When variables leave the window, their linearized information is summarized by the Schur complement

$$\boldsymbol{H}_{rr}^m = \boldsymbol{H}_{rr} - \boldsymbol{H}_{rd} \boldsymbol{H}_{dd}^{-1} \boldsymbol{H}_{dr}, \qquad \boldsymbol{g}_r^m = \boldsymbol{g}_r - \boldsymbol{H}_{rd} \boldsymbol{H}_{dd}^{-1} \boldsymbol{g}_d. \tag{7}$$

This marginal factor is the memory of the removed states. Full batch FGO avoids this truncation by keeping the selected trajectory active until the final solution, which makes it the most suitable variant here for reference quality post processing.

Fig. 2. The trajectory of the Mecklenburg Vorpommern evaluation segment and the measurement platform.

III. EXPERIMENTAL SETUP

A. Dataset and Controlled Comparison

The data set is an open sky kinematic GNSS trajectory from the Baltic Sea area. It contains approximately 6000 epochs sampled at 2 Hz, corresponding to about 50 minutes. The setup is single antenna and GNSS only. The comparison uses the same preprocessing, satellite selection, observations, stochastic model, process model, and float ambiguity handling for all algorithms. The evaluated variants are ESKF, ESKF followed by RTS smoothing, sliding window FGO with $W = 2$, sliding window FGO with $W = 30$, and full batch FGO.

Fig. 2 shows the experimental trajectory and the measurement platform. The evaluation metric is RMSE in the local east, north, up frame, together with horizontal and 3D RMSE. The same reference and the same epoch set are used for all estimators. The results should be interpreted as a controlled comparison of estimator performance under the same PPP float setup, not as the final limit of GNSS FGO in degraded environments.

The RMSE reference is the post processed Canadian Spatial Reference System Precise Point Positioning solution, CSRS-PPP, provided by the Canadian Geodetic Survey of Natural Resources Canada [25]. It is an external PPP reference and not an independent sensor ground truth such as a surveyed INS trajectory. Therefore, the reported RMSE should be read as agreement with a high quality PPP reference generated outside the tested estimators. Banville et al. describe the ambiguity resolved CSRS-PPP service and report $\sim$1 centimeter level accurate horizontal accuracy in less than one hour for both static and kinematic processing [26].

All runs use the same PPP float settings with receiver clock random walk 0.1, code noise scale 1, and robust parameter $\kappa = 1.345$. The robust setting corresponds to Huber style residual reweighting [21]. The FGO implementation uses GTSAM iSAM2. The two epoch FGO run uses sliding

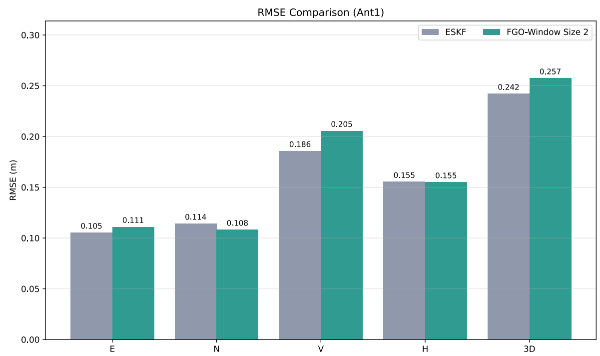

Fig. 3. RMSE confirms that sliding window FGO with $W = 2$ remains close to the ESKF baseline.

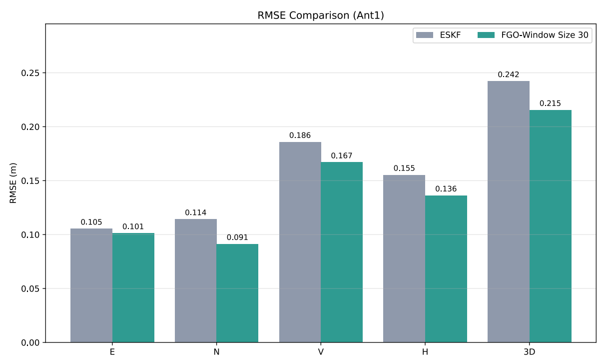

Fig. 4. FGO with window size 30 improves horizontal and vertical RMSE relative to ESKF.

mode, stride one, separated ambiguity variables, and exact marginalization when states leave the window. The full batch run uses the full window size and the stable full batch profile.

B. Two Epoch and 30 Epoch Graphs

Fig. 3 verifies the expected near parity between the two epoch graph and the ESKF. The horizontal RMSE is identical at 0.155 m, while the 3D RMSE is 0.242 m for ESKF and 0.257 m for the two epoch graph. This result is important because it prevents over interpretation of FGO. With only two epochs active, the graph has little more temporal information than the forward filter. If it had produced a dramatic improvement, that would suggest that a model, weighting, or preprocessing difference had entered the comparison.

Fig. 4 shows that a longer graph memory changes the behavior. The 30 epoch graph reduces horizontal RMSE from 0.155 m to 0.136 m and 3D RMSE from 0.242 m to 0.215 m relative to ESKF. The vertical RMSE also improves from 0.186 m to 0.167 m. In open sky these gains are moderate, but they are consistent with the role of FGO: residual biases, ambiguity coupling, and temporally correlated effects can be distributed over more connected states rather than being absorbed locally.

Table I provides the numerical summary. The main trend is memory dependent. A two epoch graph behaves almost like a filter, while a 30 epoch graph begins to use the additional temporal context.

TABLE I

RMSE COMPARISON FOR THE FILTER AND SLIDING GRAPH VARIANTS.

Metric	ESKF	FGO $W = 30$
East	0.105	**0.101**
North	0.114	**0.091**
Up	0.186	**0.167**
Horizontal	0.155	**0.136**
3D	0.242	**0.215**

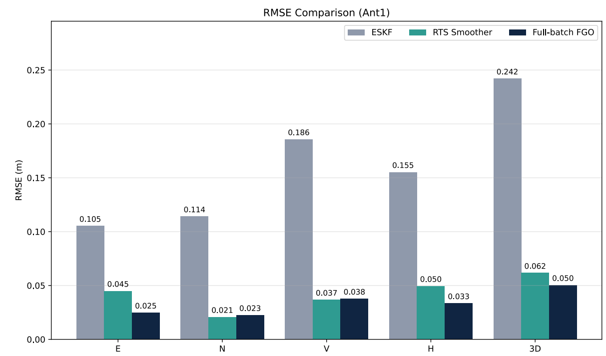

Fig. 5. Full batch FGO delivers the strongest overall post processed solution.

TABLE II

RMSE COMPARISON FOR POST PROCESSED ESTIMATORS.

Metric	ESKF	RTS	Full FGO
East	0.105	0.045	**0.025**
North	0.114	0.021	**0.023**
Up	0.186	0.037	**0.038**
Horizontal	0.155	0.050	**0.033**
3D	0.242	0.062	**0.050**

C. Full Batch Reference Trajectory

Fig. 5 compares ESKF, RTS smoothing, and full batch FGO. The full batch graph gives the lowest horizontal and 3D RMSE in the tested set. The horizontal RMSE decreases from 0.155 m for ESKF to 0.050 m for RTS and 0.033 m for full batch FGO. The 3D RMSE decreases from 0.242 m for ESKF to 0.062 m for RTS and 0.050 m for full batch FGO. RTS is already a strong offline baseline, so the additional improvement of full batch FGO indicates that direct joint optimization of all active nonlinear residuals provides a more consistent post processed trajectory than smoothing the already filtered solution.

Table II supports the reference trajectory interpretation. Full batch FGO is not the preferred online estimator because of its computational load, but it is the strongest candidate among the tested methods for producing an offline trajectory against which lower latency methods can be assessed.

D. Runtime and Scalability

Fig. 6 addresses the engineering feasibility question. The short window graph remains close to the recursive

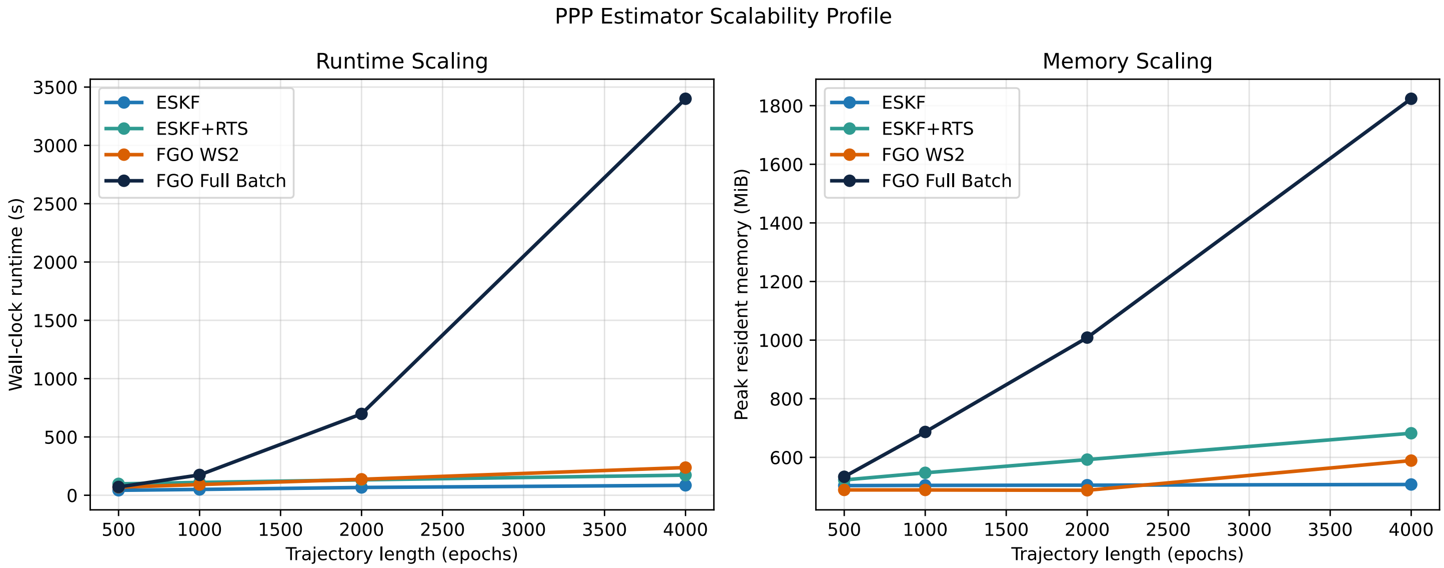

Fig. 6. Scalability profile with full batch FGO included. Runtime and peak memory grow slowly for the recursive ESKF, ESKF plus RTS, and the two epoch sliding graph. Full batch FGO has the largest runtime and memory footprint because all epoch states, nuisance states, ambiguities, and carrier phase factors remain active until the final solution.

estimators because the active problem size is bounded. The full batch graph is different, it is useful as an offline reference generator, but the cost grows with trajectory length. This is the reason why the practical deployment choice is not simply ESKF or FGO. For online use, ESKF or a short sliding graph is preferable and for post processed reference generation, full batch FGO is justified when the extra runtime and memory are acceptable.

E. Interpretation for Harsh Environments

The current experiment does not claim to exploit the full potential of FGO. The data set is open sky, no IAR is applied, no PPP RTK corrections are used, and no additional sensors are fused. This is a conservative single antenna PPP float baseline. The fact that the full batch graph still improves the post processed trajectory is therefore encouraging: it suggests that graph memory alone can improve consistency even before stronger mechanisms are added.

The expected larger benefit in challenging urban or degraded GNSS environments comes from three sources. First, such environments produce residuals that are not well represented by independent white Gaussian noise. Graphs can incorporate robust loss functions, switchable factors, and temporal consistency constraints. Second, carrier phase tracking may be intermittent, but when it remains continuous, ambiguity states connect many epochs and can constrain the trajectory more strongly than code alone. Third, PPP RTK and IAR can convert reliable float ambiguity information into stronger constraints. The literature supports this direction: Li et al. show PPP RTK FGO benefits from marginalization based ambiguity propagation and a second optimization with ambiguity resolved solutions and TDCP factors [13]; Wang et al. show that ambiguity propagation is decisive in deep urban RTK [14]; Xiao et al. show that PPP ambiguity resolution can be expressed with explicit graph factors and improves PPP accuracy after fixing [15]. Thus the present paper should be viewed as the baseline layer of a future high precision GNSS trajectory generation framework.

IV. CONCLUSIONS

This paper presented a comparison of ESKF, RTS smoothing, sliding window FGO, and full batch FGO for precise GNSS trajectory generation. All estimators used the same single antenna, undifferenced multi frequency PPP float model and the same implementation settings. The goal was to evaluate which estimator provides the highest post processed precision from the available GNSS data and to identify the practical tradeoffs of FGO.

The results clarify three regimes. A two epoch graph behaves close to the ESKF, which confirms that short graph memory alone should not be expected to produce large gains in clean open sky. A 30 epoch graph gives measurable improvement because more temporal context is available to distribute residual biases, ambiguity coupling, and weak time correlations. Full batch FGO gives the strongest post processed trajectory, reducing horizontal RMSE to 0.033 m compared with 0.155 m for ESKF and 0.050 m for RTS smoothing.

The main practical conclusion is that full batch FGO is a useful reference quality trajectory generation tool. It is not proposed as the low latency estimator for online navigation. Instead, it can generate a consistent offline trajectory for benchmarking filters, tuning sliding graphs, evaluating sensor fusion algorithms, and preparing reference data for later experiments. The scalability profile supports this interpretation: short window estimators remain practical for real time use, while full batch FGO should be treated as an offline post processing tool.

The current open sky result is intentionally conservative. It does not use IAR, PPP RTK corrections, switchable factors tuned for degraded environments, or additional sensors. The next step is to apply the same framework in scenarios where multipath, blockage, cycle slips, and non Gaussian residuals are more prominent. In those cases, the graph is expected to become more valuable when combined with robust residual design, validated ambiguity resolution, PPP RTK corrections, TDCP or Doppler constraints, and possibly IMU or perception factors.

References

[1] R. G. Brown and P. Y. C. Hwang, *Introduction to Random Signals and Applied Kalman Filtering: With MATLAB Exercises*, 4th ed. Hoboken, NJ, USA: Wiley, 2012.

[2] P. D. Groves, *Principles of GNSS, Inertial, and Multisensor Integrated Navigation Systems*, 2nd ed. Boston, MA, USA: Artech House, 2013.

[3] H. E. Rauch, F. Tung, and C. T. Striebel, "Maximum likelihood estimates of linear dynamic systems," *AIAA Journal*, vol. 3, no. 8, pp. 1445–1450, 1965, doi: 10.2514/3.3166.

[4] F. Dellaert and M. Kaess, "Factor graphs for robot perception," *Foundations and Trends in Robotics*, vol. 6, no. 1–2, pp. 1–139, 2017, doi: 10.1561/2300000043.

[5] M. Kaess, H. Johannsson, R. Roberts, V. Ila, J. J. Leonard, and F. Dellaert, "iSAM2: Incremental smoothing and mapping using the Bayes tree," *The International Journal of Robotics Research*, vol. 31, no. 2, pp. 216–235, 2012, doi: 10.1177/0278364911430419.

[6] T. D. Barfoot, *State Estimation for Robotics*. Cambridge, U.K.: Cambridge University Press, 2017.

[7] C. Taylor and J. N. Gross, "Factor graphs for navigation applications: A tutorial," *NAVIGATION: Journal of the Institute of Navigation*, vol. 71, no. 3, 2024, doi: 10.33012/navi.653.

[8] J. F. Zumberge, M. B. Heflin, D. C. Jefferson, M. M. Watkins, and F. H. Webb, "Precise point positioning for the efficient and robust analysis of GPS data from large networks," *Journal of Geophysical Research: Solid Earth*, vol. 102, no. B3, pp. 5005–5017, 1997, doi: 10.1029/96JB03860.

[9] P. J. G. Teunissen and O. Montenbruck, Eds., *Springer Handbook of Global Navigation Satellite Systems*. Cham, Switzerland: Springer, 2017, doi: 10.1007/978-3-319-42928-1.

[10] W. Wen and L.-T. Hsu, "Towards robust GNSS positioning and real time kinematic using factor graph optimization," in *Proc. IEEE International Conference on Robotics and Automation (ICRA)*, Xi'an, China, 2021, pp. 5884–5890, doi: 10.1109/ICRA48506.2021.9562037.

[11] W. Wen, T. Pfeifer, X. Bai, and L.-T. Hsu, "Factor graph optimization for GNSS/INS integration: A comparison with the extended Kalman filter," *NAVIGATION: Journal of the Institute of Navigation*, vol. 68, no. 2, pp. 315–331, 2021, doi: 10.1002/navi.421.

[12] X. Bai, W. Wen, and L.-T. Hsu, "Time correlated window carrier phase aided GNSS positioning using factor graph optimization for urban positioning," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 58, no. 4, pp. 3370–3384, 2022.

[13] X. Li, X. Li, X. Wang, H. Chang, Y. Tan, and Z. Shen, "Factor graph based PPP RTK for accurate and robust positioning in urban environments," *Journal of Geodesy*, vol. 98, art. 21, 2024, doi: 10.1007/s00190-024-01828-3.

[14] X. Wang, X. Li, Z. Shen, X. Li, Y. Zhou, and H. Chang, "Factor graph optimization based multi GNSS real time kinematic system for robust and precise positioning in urban canyons," *GPS Solutions*, vol. 27, art. 200, 2023, doi: 10.1007/s10291-023-01538-x.

[15] G. Xiao, C. Yang, H. Wei, Z. Xiao, P. Zhou, P. Li, Q. Dai, B. Zhang, and C. Yu, "PPP ambiguity resolution based on factor graph optimization," *GPS Solutions*, vol. 28, art. 178, 2024, doi: 10.1007/s10291-024-01715-6.

[16] H. Gao, H. Li, H. Huo, and C. Yang, "Robust GNSS real time kinematic with ambiguity resolution in factor graph optimization," in *Proc. International Technical Meeting of The Institute of Navigation*, 2022, pp. 835–843, doi: 10.33012/2022.18190.

[17] T. Dome, T. Russell, M. Ortiz Rejon, Y. Zheng, E. Benedetti, T. Li, M. Sun, and I. Petrunin, "Comparative analysis of factor graph models for carrier phase based precision navigation," *Engineering Proceedings*, 126(1), 11, 2026, doi: 10.3390/engproc2026126011.

[18] D. Medina, H. Li, J. Vilà-Valls, and P. Closas, "On robust statistics for GNSS single point positioning," in *Proc. IEEE Intelligent Transportation Systems Conference (ITSC)*, 2019, pp. 3281–3287, doi: 10.1109/ITSC.2019.8917484.

[19] A. Bellés, D. A. Medina, P. Chauchat, and J. Vilà-Valls, "Robust PPP-AR Using M-Estimators for Multi-Fault Scenarios," in *Proc. IEEE Aerospace Conference*, 2022, pp. 1–9, doi: 10.1109/AERO53065.2022.9843300.

[20] A. Bellés and D. A. Medina, "Robust PPP AR using M estimators for multi fault scenarios," in *Proc. IEEE/ION Position, Location and Navigation Symposium (PLANS)*, 2025, pp. 251–257, doi: 10.1109/PLANS61210.2025.11028258.

[21] P. J. Huber, "Robust estimation of a location parameter," *The Annals of Mathematical Statistics*, vol. 35, no. 1, pp. 73–101, 1964, doi: 10.1214/aoms/1177703732.

[22] T. Suzuki, "First Place Award Winner of the Smartphone Decimeter Challenge: Global Optimization of Position and Velocity by Factor Graph Optimization," in *Proc. 34th International Technical Meeting of the Satellite Division of The Institute of Navigation (ION GNSS+ 2021)*, St. Louis, MO, USA, 2021, pp. 2974–2985, doi: 10.33012/2021.18109.

[23] T. Suzuki, "Second Place Winner of the Smartphone Decimeter Challenge: An Open-Source Factor Graph Optimization Package for GNSS and IMU Integration in Smartphones," in *Proc. 37th International Technical Meeting of the Satellite Division of The Institute of Navigation (ION GNSS+ 2024)*, Baltimore, MD, USA, 2024, pp. 2703–2713, doi: 10.33012/2024.19923.

[24] H. Zhang, C.-C. Chen, H. Vallery, and T. D. Barfoot, "GNSS/multisensor Fusion Using Continuous-Time Factor Graph Optimization for Robust Localization," *IEEE Transactions on Robotics*, vol. 40, pp. 4003–4023, 2024, doi: 10.1109/TRO.2024.3443699.

[25] Natural Resources Canada, Canadian Geodetic Survey, "Precise Point Positioning," CSRS-PPP online service, accessed May 22, 2026. [Online]. Available: `https://webapp.csrs-scrs.nrcan-rncan.gc.ca/geod/tools-outils/ppp.php?locale=en`

[26] S. Banville, E. Hassen, P. Lamothe, J. Farinaccio, B. Donahue, Y. Mireault, M. A. Goudarzi, P. Collins, R. Ghoddousi-Fard, and O. Kamali, "Enabling ambiguity resolution in CSRS-PPP," *NAVIGATION: Journal of the Institute of Navigation*, vol. 68, no. 2, pp. 433–451, 2021, doi: 10.1002/navi.423.