KF-GINS: An Open-Sourced Software for GNSS/INS Integrated Navigation

Xiaoji Niu[1,2], Liqiang Wang[1,4], Qijin Chen[1,3], Hailiang Tang[1,2], Quan Zhang[1,2] and Tisheng Zhang[1,3*]

[1] GNSS Research Center, Wuhan University, Wuhan 430079, China
[2] Hubei Technology Innovation Center for Spatiotemporal Information and Positioning Navigation, Wuhan, 430079, China
[3] Hubei Luojia Laboratory, Wuhan, 430079, China
[4] Electronic Information School, Wuhan University, 430079, China
* Corresponding author; email: zts@whu.edu.cn

Abstract

The loosely-coupled system is the most popular architecture in Global Navigation Satellite System (GNSS) / Inertial Navigation System (INS) integration, offering a starting point for beginners entering the field of positioning and navigation. Despite well-established theoretical foundations, developing fully functional integrated navigation algorithms remains challenging, especially for early-career researchers. To meet this challenge, we have developed and open-sourced a C++-based GNSS/INS data processing software, named KF-GINS, which uses Extended Kalman Filter (EKF) to implement loosely-coupled GNSS/INS integration. Accompanied by our previously released video courses, tutorial documents, and result analysis scripts, KF-GINS serves as a comprehensive learning resource and a dependable research platform for those new to GNSS/INS integration. To facilitate algorithm development and exploration, we further release KF-GINS-Matlab, a Matlab version counterpart to KF-GINS. Sharing identical architecture and core algorithms with KF-GINS, KF-GINS-Matlab effectively bridges the gap between algorithm research and engineering implementation. Experimental data processing results and analysis confirm the algorithm's correctness of KF-GINS. Besides, a comprehensive evaluation of the navigation accuracy indicates that KF-GINS achieves comparable performance in terms of positioning accuracy with the renowned commercial software NovAtel Inertial Explorer (IE) within the loosely-coupled framework.

Keywords: GNSS/INS integrated navigation, loosely-coupled integration, Kalman filter, open-sourced software.

Introduction

Continuous and accurate localization remains crucial for autonomous vehicles and unmanned systems in complex environments. Among positioning technologies, the Global Navigation Satellite System (GNSS) provides real-time, all-weather, absolute positioning results, with drift-free errors over time (Teunissen and Montenbruck 2017; Fan et al. 2023). However, GNSS accuracy degrades or becomes unavailable when signals are blocked. Conversely, the Inertial Navigation System (INS), a self-contained dead-reckoning system, derives position and attitude from acceleration and angular velocity measured by the Inertial Measurement Unit (IMU), operating independently of external conditions (El-Sheimy and Youssef 2020). As a relative positioning method, however, INS suffers from rapidly accumulating errors over time(Wang et al. 2021). Given their complementarity, GNSS and INS are often integrated into widely used GNSS/INS navigation systems.

GNSS/INS integration can be categorized into loosely-coupled, tightly-coupled, and deeply-coupled architectures based on the level of information fusion (Niu et al. 2023; Boguspayev et al. 2023). Among these, the loosely-coupled approach integrates GNSS positioning results with INS navigation states through a navigation filter, such as an Extended Kalman Filter (EKF), to optimally estimate position and attitude. Benefiting from its simple structure and strong robustness, the loosely-coupled GNSS/INS algorithm is widely used in applications like railway track positioning, autonomous driving, unmanned aerial vehicles, and consumer devices. Furthermore, as a fundamental component of integrated navigation, it supports algorithm development in wheeled vehicle positioning algorithms (Chen et al. 2020b; Ouyang et al. 2021), state initialization algorithms (Chen et al. 2020a), and multi-sensor fusion algorithms (Chiang et al. 2020).

Although GNSS/INS loosely-coupled algorithms are well-established, their implementation still requires a solid understanding of inertial navigation algorithms, IMU error modelling, and Kalman filter design, making the development of a complete integrated navigation algorithm challenging for beginners. Several excellent open-sourced GNSS processing software packages, such as RTKLIB (Takasu and Yasuda 2009), PPP-AR (Geng et al. 2019), and GREAT-UPD (Li et al. 2021), have significantly advanced the GNSS research community. However, open-source solutions specifically designed for GNSS/INS integrated navigation remain scarce. Existing platforms, such as INS Toolkit (2023), PSINS (Yan 2024) and GINAV (Chen et al. 2021), are primarily Matlab-based, limit engineering applications and code migration. Although a C-language version of PSINS exists, its limited library and code architecture constrain its extensibility. Consequently, there remains a pressing need for an open-sourced, C++-based GNSS/INS loosely-coupled software to advance engineering applications of GNSS/INS integration. A corresponding Matlab version could further support algorithmic research and engineering practice. Furthermore, for newcomers entering the field of GNSS/INS loosely-coupled integration, access to structured learning materials, including video tutorials, algorithm documentation, and well-annotated code, is just as crucial as open-source software. However, such resources are still limited, posing a barrier to entry.

To bridge these gaps, we have open-sourced two similarly structured EKF-based GNSS/INS loosely-coupled navigation algorithms, including C++ version KF-GINS[1] and Matlab version KF-GINS-Matlab[2], which have received 900+ stars on GitHub. In addition, supporting video courses3 (Niu and Chen 2021) and algorithm tutorials4 (Niu and Chen 2022) have also been made publicly available. The key contributions are summarized as follows:

- We have open-sourced KF-GINS, a C++ version GNSS/INS loosely-coupled navigation algorithm that focuses on inertial navigation and loosely-coupled integration algorithms. Accompanied by our publicly available video courses and tutorial documents, KF-GINS serves as both an entry-level learning resource for new researchers in GNSS/INS integration and a foundational research platform for GNSS/INS integration algorithms.
- To facilitate software debugging and algorithm verification for newcomers, we have also released KF-GINS-Matlab, a Matlab version GNSS/INS loosely-coupled algorithm. KF-GINS-Matlab shares the same core algorithms and software architecture as KF-GINS, effectively bridging the gap between algorithm research (MATLAB) and engineering implementation (C++).
- A comprehensive analysis and an accuracy evaluation of the proposed software have been conducted. The result analysis confirms the algorithm's correctness in KF-GINS and demonstrates positioning accuracy comparable to that of the commercial-grade software, Inertial Explorer (IE).

The following sections present the software design and implementation, including an overview and processing flow. Experimental results and analyses are then provided to validate the solution, followed by conclusions and future directions.

Software Design and Implementation

Software Overview

KF-GINS and KF-GINS-Matlab implement a loosely-coupled GNSS/INS integration algorithm based on EKF, which is in fact an error-state KF. The software architecture is shown in Fig. 1. KF-GINS takes IMU data and GNSS position data as inputs, producing navigation state (position, velocity, and attitude), IMU error estimates, and the standard deviations (STD) of the navigation states. KF-GINS-Matlab extends KF-GINS by integrating GNSS velocity and odometer (ODO) / non-holonomic constraint (NHC) measurements, with optional inputs for GNSS velocity and ODO speed data.

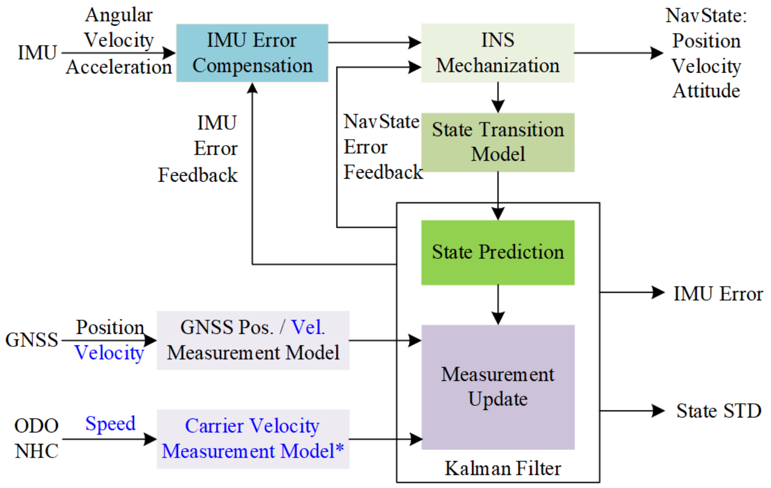


Fig.1 System architecture of KF-GINS and KF-GINS-Matlab. Components with blue words are exclusive to KF-GINS-Matlab. *The ODO/NHC measurement function is not fully implemented.

During the software execution, IMU measurement errors are first compensated before performing INS mechanization to obtain position, velocity, and attitude information. The resulting INS navigation states are used to construct the state transition model for the Kalman filter prediction step. When GNSS positions are available, they are used to construct the measurement matrix and measurement vector for the Kalman filter update. After measurement updates, the estimated error states are fed back to correct the system navigation states and IMU error estimation. Additionally, KF-GINS-Matlab supports GNSS velocity and ODO/NHC updates, where the ODO/NHC measurement model is left blank as a learning exercise for beginners. The subsequent sections focus on the common core of KF-GINS and KF-GINS-Matlab, using KF-GINS as the representative example.

[1]Online available at: https://github.com/i2Nav-WHU/KF-GINS
[2]Online available at: https://github.com/i2Nav-WHU/KF-GINS-Matlab
[3]Online available at: https://www.bilibili.com/video/BV1na411Z7rQ
[4]Online available at: http://i2nav.com/index/newList_zw?newskind_id=13a8654e060c40c69e5f3d4c13069078

KF-GINS considers IMU errors, including biases, scale factors, and measurement noise. IMU biases and scale factors are corrected in the IMU error compensation stage (Shin 2005). The INS mechanization employs a second-order integration method with compensation for coning effects for attitude, rotational effects and sculling effects for velocity (Savage 1998a; Savage 1998b). The system error state model consists of 21 states, including position errors, velocity errors, attitude errors, IMU bias errors, and IMU scale factor errors. The position errors and velocity errors are modelled in the navigation frame, attitude errors are represented using the Phi-angle model, while IMU bias errors and scale factor errors are modelled as first-order Gauss-Markov processes (Shin 2005; Niu et al. 2015). Detailed algorithm documentation is provided in the supplementary file.

Software Process Flow

The main processing flow of KF-GINS is illustrated in Fig. 2. First, configuration parameters are loaded from a configuration file. Then, the loosely-coupled navigation system is initialized using these parameters. The software then iteratively reads test data in time order. The navigation solutions are computed using the core loose-coupled algorithm, and the results are saved until reaching the end time.

As shown in Fig. 3, the core of the loosely-coupled processing consists of three main components: INS propagation, KF system state prediction, and GNSS update & feedback. In Fig. 3, the term 'new data' refers to the newest IMU and GNSS measurements. When a new IMU measurement is received, the system first performs INS propagation, which includes IMU error compensation and INS mechanization. Subsequently, the system performs state prediction, during which the state transition matrix and system noise matrix are constructed for the Kalman filter prediction step. If a GNSS measurement is available, the GNSS update and feedback are executed, including innovation computation, Kalman updates, and error state feedback. After the feedback step, the Kalman filter error state is reset to zero.

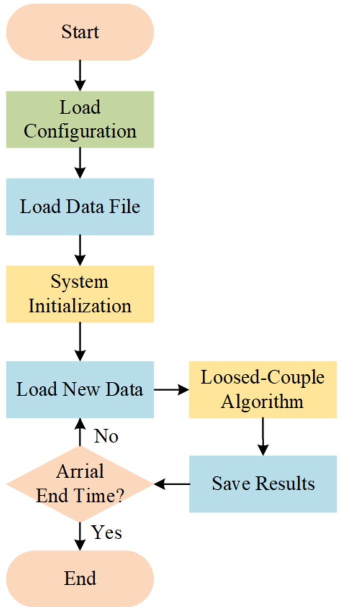


Fig.2 Main processing flow of KF-GINS

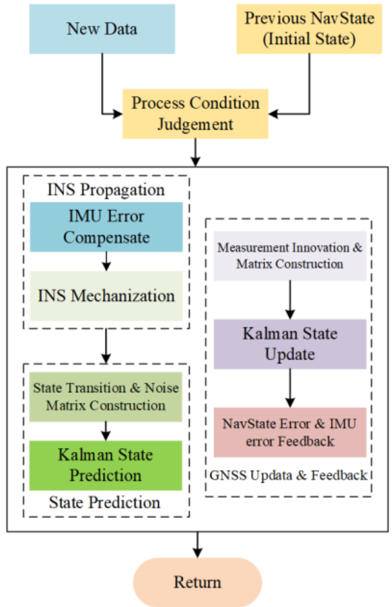


Fig.3 Core of the loosely-coupled algorithm.

IMU and GNSS data have different sampling rates and misaligned timestamps. The core loosely-coupled algorithm adapts its processing strategy based on their timing relationship, as shown in Fig. 4, with corresponding strategies in Fig. 5. When the GNSS timestamp falls outside the interval between the previous and new IMU data (Condition #1), only the new IMU data is processed (INS propagation and state prediction). If the GNSS timestamp is within a predefined threshold of the previous IMU timestamp (Condition #2), GNSS updates and feedback are performed first, followed by processing the new IMU data. If it is within the threshold of the new IMU timestamp (Condition #3), the new IMU data is processed first, followed by the GNSS update. If the GNSS timestamp falls between two IMU samples and is not close to either (Condition #4), the new IMU data is split into two segments under a constant acceleration and angular velocity assumption: the first segment propagates the state up to the GNSS time (after which GNSS update is applied), and the second segment propagates the state to the new IMU time. For low-speed platform or poor GNSS observations, the processing strategy may rely solely on Conditions #2 and #3.

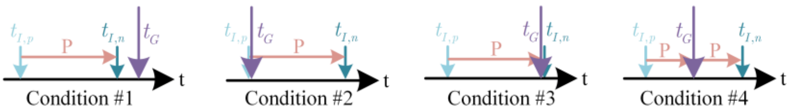


Fig.4 Sampling time relationship between IMU and GNSS data. $t_{I,p}$, $t_{I,n}$, and t_G denote timestamps of previous IMU, new IMU, and GNSS data; 'P' denotes INS propagation and state prediction.

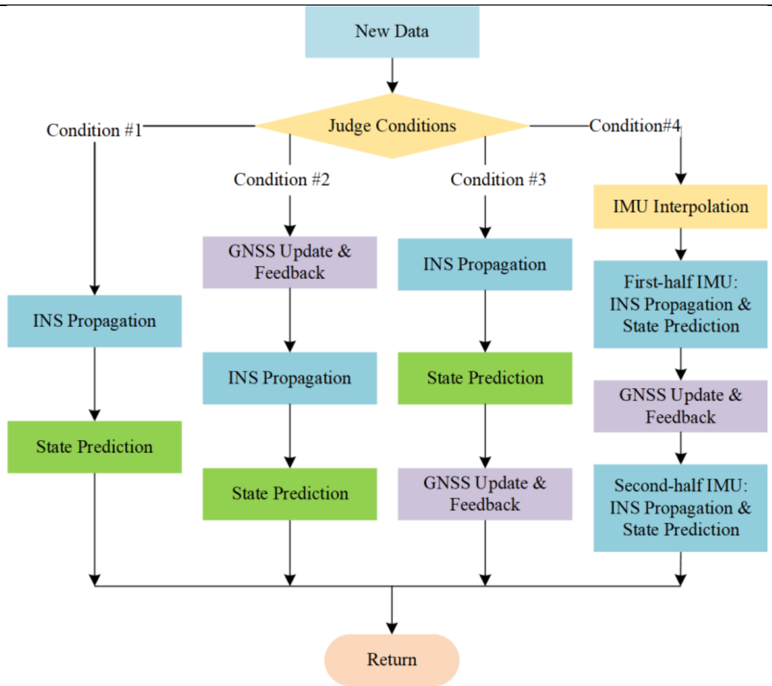


Fig.5 Processing cases for the loosely-coupled algorithm.

KF-GINS-Matlab also includes an ODO/NHC update framework. As ODO velocity measurements are typically high data rate (e.g. same as the IMU data rate) and subject to large quantization noise, KF-GINS-Matlab downsamples ODO data based on a predefined update frequency (e.g. 1Hz). Linear extrapolation is employed to estimate ODO velocity at the new IMU timestamp, enabling time alignment and reducing quantization errors in the ODO/NHC measurement update (Wang et al. 2022).

Algorithm Verification

Field experiments were conducted to validate the proposed software, including algorithm correctness and positioning accuracy. As KF-GINS and KF-GINS-Matlab share the same algorithm design and software architecture, only KF-GINS was used in the following experiments.

Correctness Verification

The field experiment provides a set of open-sky vehicle test data, which includes centimeter-level GNSS RTK positioning results and IMU measurements from a navigation-grade IMU, Leador-A15. The main specifications of the IMU are listed in Table 1. The processing results are illustrated in Fig. 6.

Table 1 Main specifications of the tested IMU (Leador-A15)

IMU	Parameters	Values	Units
Gyroscope	Angular random walk	0.003	deg/sqrt(h)
	Bias instability	0.027	deg/h
	Scale factor instability	300	ppm
Accelerometer	Velocity random walk	0.03	m/s/sqrt(h)
	Bias instability	15	mGal
	Scale factor instability	300	ppm
Correlation time		4	hour

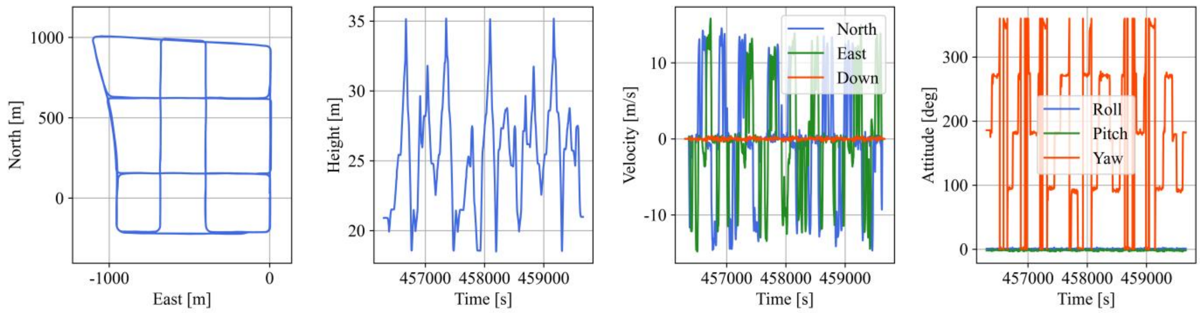


Fig. 6 KF-GINS Processing results: trajectory, height, velocity, and attitude.

The estimated IMU biases and scale factors are shown in Fig. 7. The gyroscope errors fluctuate near zero, consistent with navigation-grade IMU characteristics. The accelerometer errors exhibit a clear convergence trend after a period of vehicle manoeuvres. Due to the limited dynamics of the land vehicle, the Z-axis accelerometer bias and scale factor are strongly coupled to each other (Zhang et al. 2024), resulting in minimal fluctuation in the Z-axis bias. Notably, once converged, all estimated biases and scale factors remain within three standard deviations of the IMU noises, aligning with the assumptions on the predefined IMU error model in KF-GINS.

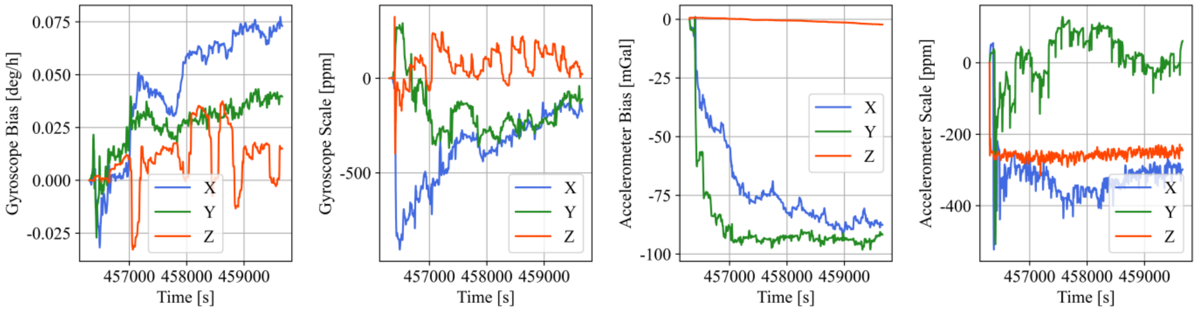
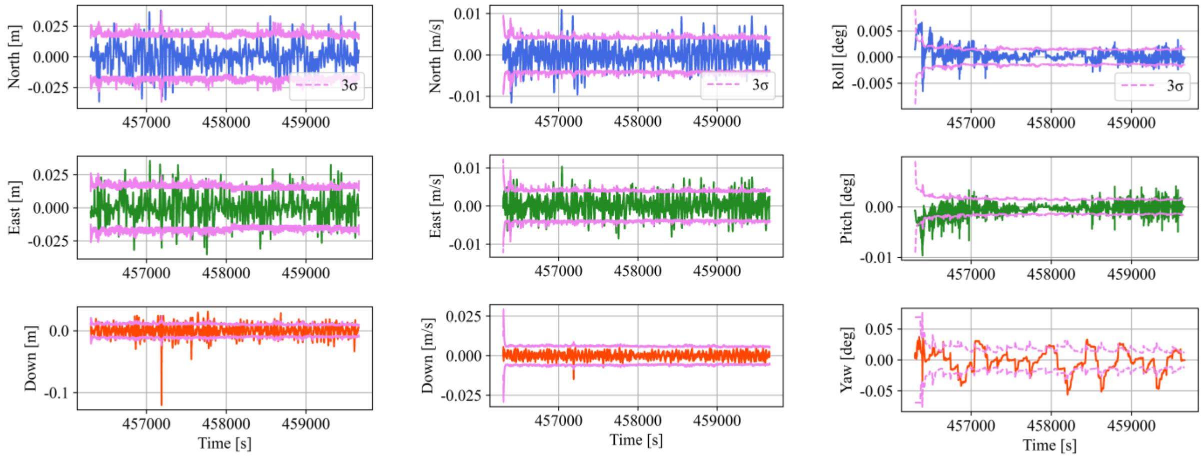


Fig. 7 The estimated IMU errors obtained from KF-GINS.

The navigation state errors and corresponding standard deviations output by KF-GINS are shown in Fig. 8, where the pink curve represents the three-sigma bounds. Except for the spike in the height error curve caused by a GNSS outlier, the state errors demonstrate that KF-GINS outputs accurate estimates of position, velocity, and attitude. Moreover, navigation state standard deviations converge quickly after initialization. The variations in position, velocity, and attitude errors remain within the three-sigma bounds, indicating strong consistency between the estimated uncertainties and the actual navigation errors. In summary, the estimated IMU errors, positioning errors, and the corresponding standard deviations validate the correctness and effectiveness of the KF-GINS algorithm.



(a) Position errors (b) Velocity errors (c) Attitude errors

Fig. 8 Navigation states errors with three-sigma bounds (pink lines).

(a) Test A (b) Test B (c) Test C

Fig. 9 Test trajectories (green points) and GNSS positioning results (red points) of the test datasets.

Accuracy Evaluation

To evaluate the positioning accuracy of KF-GINS, we conducted side-by-side comparisons with Inertial Explorer (IE) (NovAtel 2024), a widely used commercial GNSS/INS data processing software. The evaluation was carried out using three sets of open-sky vehicle test data, collected with the navigation-grade IMU (Leador-A15) and corresponding GNSS data. To assess the GNSS/INS navigation performance, we manually introduced GNSS outages of 60 seconds every 180 seconds after system convergence (Niu et al. 2006). The outaged GNSS data served as inputs for the loosely-coupled integration in both IE and KF-GINS. The trajectories and GNSS positioning results for the three test scenarios are illustrated in Fig. 9.

To simulate more outages, we shifted the start times by 60 and 120 seconds and repeated the evaluation procedure. As shown in Fig. 10, we present the horizontal positioning errors of IE and KF-GINS of such outage tests in the first dataset. As expected, both systems showed increased positioning errors during GNSS outages but maintained comparable accuracy throughout. Please note that IE applies zero-velocity updates (ZUPT) by default in its loosely-coupled mode, thus utilizing more aiding information than KF-GINS.

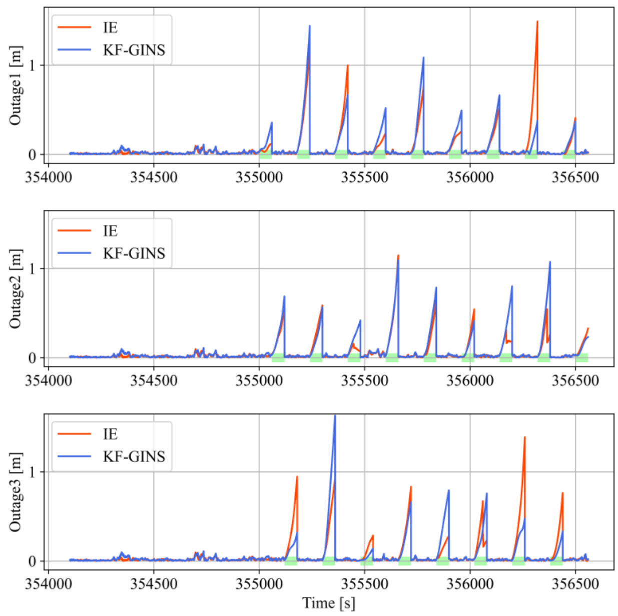


Fig.10 Horizontal positioning errors of IE and KF-GINS in the first dataset. Green zones denote GNSS outages.

Table. 2 Position and attitude drift errors (RMS) during 60s GNSS outages across three field test datasets

Test Data	Software	Position error [m]			Attitude error [deg]		
		Horizon	Height	3D	Roll	Pitch	Heading
A	NovAtel IE	0.741	0.383	0.835	0.001	0.002	0.012
	KF-GINS	0.737	0.287	0.791	0.001	0.001	0.018
B	NovAtel IE	0.743	0.638	0.979	0.002	0.002	0.012
	KF-GINS	0.662	0.899	1.116	0.001	0.001	0.019
C	NovAtel IE	0.663	0.258	0.711	0.002	0.002	0.012
	KF-GINS	0.524	0.379	0.646	0.001	0.001	0.022
ALL	NovAtel IE	**0.717**	**0.455**	**0.849**	**0.002**	**0.002**	**0.012**
	KF-GINS	**0.647**	**0.587**	**0.874**	**0.001**	**0.001**	**0.020**

Table 2 summarizes the position and attitude drifts during simulated 60s GNSS outages across all three field test datasets. Concretely, the maximum position and attitude drift errors in each outage were recorded, and the root-mean-square (RMS) of these maximum drifts was computed. Due to varying test conditions, height errors in Group B are worse than those in Groups A and C. The weak observability of the heading angle estimation under regular vehicle motions results in larger heading errors compared to roll and pitch angles across all tests. Differences in IMU modelling and ZUPT use in IE also cause discrepancies in horizontal and vertical positioning errors. Nonetheless, their overall 3-dimensional (3D) position errors are similar, indicating their comparable positioning performance. Furthermore, the weak heading observability makes heading accuracy highly sensitive to random factors such as parameter tuning, leading to noticeable differences in heading errors between IE and KF-GINS. In contrast, the differences in roll and pitch angles are relatively small, indicating comparable attitude accuracy across both systems. In conclusion, results from multiple field tests confirm that KF-GINS achieves positioning and attitude performance comparable to the commercial software NovAtel IE.

Conclusion

We have open-sourced an EKF-based loosely-coupled GNSS/INS navigation software, including C++ version KF-GINS and Matlab version KF-GINS-Matlab. Supported by publicly available video courses, tutorial documents, and result analysis scripts, these tools offer an accessible research platform for GNSS/INS integration, bridging algorithm research (MATLAB) and engineering implementation (C++). The results and analysis verified the functionality, accuracy and reliability of KF-GINS. Accuracy evaluation further indicates that KF-GINS achieves comparable performance to the renowned commercial-grade software NovAtel IE in the loosely-coupled framework.

The source codes and result analysis scripts for both KF-GINS and KF-GINS-Matlab are available at: https://github.com/i2Nav-WHU/KF-GINS and https://github.com/i2Nav-WHU/KF-GINS-Matlab. In future, we plan to extend KF-GINS with additional measurement models and various initialization modes.

Appendix

Acknowledgement

This work was supported by the National Natural Science Foundation of China (42374034), Key Research and Development Program of Hubei Province (2024BAB024), Major Program of Hubei Province (2023BAA02602), and High quality development project of MIIT (2024-182).

Data Availability

The test data are available from https://github.com/i2Nav-WHU/KF-GINS/tree/main/dataset.

Reference

Boguspayev N, Akhmedov D, Raskaliyev A, Kim A, Sukhenko A (2023) A Comprehensive Review of GNSS/INS Integration Techniques for Land and Air Vehicle Applications. Applied Sciences 13(8):4819. https://doi.org/10.3390/app13084819

Chen K, Chang G, Chen C (2021) GINav: a MATLAB-based software for the data processing and analysis of a GNSS/INS integrated navigation system. GPS Solut 25(3):108. https://doi.org/10.1007/s10291-021-01144-9

Chen Q, Lin H, Guo R, Niu X (2020a) Rapid and accurate initial alignment of the low-cost MEMS IMU chip dedicated for tilted RTK receiver. GPS Solut 24(4):119. https://doi.org/10.1007/s10291-020-01032-8

Chen Q, Zhang Q, Niu X (2020b) Estimate the Pitch and Heading Mounting Angles of the IMU for Land Vehicular GNSS/INS Integrated System. IEEE Trans Intell Transport Syst :1–13. https://doi.org/10.1109/TITS.2020.2993052

Chiang K-W, Chang H-W, Li Y-H, Tsai G-J, Tseng C-L, Tien Y-C, Hsu P-C (2020) Assessment for INS/GNSS/Odometer/Barometer Integration in Loosely-Coupled and Tightly-Coupled Scheme in a GNSS-Degraded Environment. IEEE Sensors J 20(6):3057–3069. https://doi.org/10.1109/JSEN.2019.2954532

El-Sheimy N, Youssef A (2020) Inertial sensors technologies for navigation applications: state of the art and future trends. Satellite Navigation 1(1):2. https://doi.org/10.1186/s43020-019-0001-5

Fan T, Zhang T, Zhang H, Mo J, Niu X (2023) A double sideband combined tracking method for Galileo E5 AltBOC signals. Satell Navig 4(1):27. https://doi.org/10.1186/s43020-023-00117-2

Geng J, Chen X, Pan Y, Mao S, Li C, Zhou J, Zhang K (2019) PRIDE PPP-AR: an open-source software for GPS PPP ambiguity resolution. GPS Solut 23(4):91. https://doi.org/10.1007/s10291-019-0888-1

Li X, Han X, Li X, Liu G, Feng G, Wang B, Zheng H (2021) GREAT-UPD: An open-source software for uncalibrated phase delay estimation based on multi-GNSS and multi-frequency observations. GPS Solut 25(2):66. https://doi.org/10.1007/s10291-020-01070-2

Niu, Goodall C, Nassar S, El-Sheimy N (2006) An Efficient Method for Evaluating the Performance of MEMS IMUs. In: 2006 IEEE/ION Position, Location, And Navigation Symposium. IEEE, Coronado, CA, pp 766–771

Niu X, Chen Q (2021) Principles of Inertial Navigation System

Niu X, Chen Q (2022) KF Design for the loosely GNSS/INS data fusion

Niu X, Dai Y, Liu T, Chen Q, Zhang Q (2023) Feature-based GNSS positioning error consistency optimization for GNSS/INS integrated system. GPS Solut 27(2):89. https://doi.org/10.1007/s10291-023-01421-9

Niu X, Zhang Q, Gong L, Liu C, Zhang H, Shi C, Wang J, Coleman M (2015) Development and evaluation of GNSS/INS data processing software for position and orientation systems. Survey Review 47(341):87–98. https://doi.org/10.1179/1752270614Y.0000000099

NovAtel (2024) Inertial Explorer GNSS+INS post-processing

Ouyang W, Wu Y, Chen H (2021) INS/Odometer Land Navigation by Accurate Measurement Modeling and Multiple-Model Adaptive Estimation. IEEE Trans Aerosp Electron Syst 57(1):245–262. https://doi.org/10.1109/TAES.2020.3011998

Savage PG (1998a) Strapdown Inertial Navigation Integration Algorithm Design Part 1: Attitude Algorithms. Journal of Guidance, Control, and Dynamics 21(1):19–28. https://doi.org/10.2514/2.4228

Savage PG (1998b) Strapdown Inertial Navigation Integration Algorithm Design Part 2: Velocity and Position Algorithms. Journal of Guidance, Control, and Dynamics 21(2):208–221. https://doi.org/10.2514/2.4242

Shin E-H (2005) Estimation techniques for low-cost inertial navigation. Ph.D., University of Calgary

Takasu T, Yasuda A (2009) Development of the low-cost RTK-GPS receiver with an open source program package RTKLIB. International symposium on GPS/GNSS

Teunissen PJG, Montenbruck O (2017) Springer handbook of global navigation satellite systems. Springer, Cham, Switzerland

Wang L, Niu X, Zhang T, Tang H, Chen Q (2022) Accuracy and robustness of ODO/NHC measurement models for wheeled robot positioning. Measurement 201:111720. https://doi.org/10.1016/j.measurement.2022.111720

Wang L, Tang H, Zhang T, Chen Q, Shi J, Niu X (2021) Improving the Navigation Performance of the MEMS IMU Array by Precise Calibration. IEEE Sensors J 21(22):26050–26058. https://doi.org/10.1109/JSEN.2021.3118455

Yan G (2024) Precise Strapdown Inertial Navigation System (PSINS) Toolbox for MATLAB

Zhang T, Wang G, Chen Q, Tang H, Wang L, Niu X (2024) Influence Analysis of IMU Scale Factor Error in GNSS/MEMS IMU Vehicle Integrated Navigation. Journal of Geodesy and Geodynamics 44(2):134–137. https://doi.org/10.14075/j.jgg.2023.04.172

(2023) Aided Inertial Navgation Project. www.instk.org

Author Biographies

Xiaoji Niu is a Professor at GNSS Research Center, Wuhan University, China. He was conferred Ph.D. and bachelor's degrees (with honors) by the Department of Precision Instruments at Tsinghua University in 2002 and 1997, respectively. He performed post-doctoral research at the University of Calgary, Canada, and worked as a senior scientist at SiRF Technology Inc. Dr. Niu is currently leading the Integrated & Intelligent Navigation (i2Nav) group. His research interests focus on GNSS/INS integration, low-cost navigation sensor fusion, and relevant new applications. Dr. Niu has published 200+ academic papers and owns 50+ patents.

Liqiang Wang received the B.E. and M.E. degrees (with honors) from Wuhan University, China, in 2020 and 2023, respectively. He is currently pursuing a Ph.D. degree at GNSS Research Center, Wuhan University. His primary research interests include GNSS/INS integrations, visual SLAM, and multi-sensor fusion navigation systems.

Qijin Chen received the B.Eng. degree and Ph.D. degree in Geodesy and Survey Engineering from Wuhan University, Wuhan, China, in 2011 and 2016, respectively. He is currently an Associate Professor with the GNSS Research Center, Wuhan University, Wuhan, China. His research interests focus on INS with aiding and its applications in geodesy and precise surveying engineering including railway track geometry measuring and underground pipeline surveying.

Hailiang Tang received the M.E. and Ph.D. degrees from Wuhan University, China, in 2020 and 2023, respectively. He is currently a postdoctoral fellow at GNSS Research Center, Wuhan University. His research interests include autonomous robotics systems, visual and LiDAR SLAM, GNSS/INS integration, and deep learning.

Quan Zhang received the B.S. degree in Geomatics Engineering from Shandong University of Science and Technology, China, in 2009, and the Ph.D. degree from Wuhan University, China, in 2015. From 2017 to 2018, he was a Post-Doctoral Researcher with the Digital Photogrammetry Research Group (DPRG), Lyles School of Civil Engineering of Purdue University. He is currently an Associate Professor at GNSS Research Center in Wuhan University. His research interests include vehicle-mounted GNSS/INS integration technology and the integrity of multi-sensor integrated navigation for intelligent driving.

Tisheng Zhang is an Associate Professor at GNSS Research Center, Wuhan University, China. He received his B.SC. and Ph.D. degrees in Communication and Information Systems from Wuhan University, Wuhan, China, in 2008 and 2013, respectively. He was a Post-Doctor at Hong Kong Polytechnic University from 2018 to 2019. His research interests focus on the fields of GNSS receiver and multi-sensor integrated navigation.