

LIGO: A Tightly Coupled LiDAR-Inertial-GNSS Odometry Based on a Hierarchy Fusion Framework for Global Localization With Real-Time Mapping

Dongjiao He , Member, IEEE, Haotian Li , Graduate Student Member, IEEE, and Jie Yin 

Abstract—This article introduces a method for tightly fusing sensors with diverse characteristics to maximize their complementary properties, thereby surpassing the performance of individual components. Specifically, we propose a tightly coupled light detection and ranging (LiDAR)-inertial-global navigation satellite system (GNSS) odometry (LIGO) system, which synthesizes the advantages of LiDAR, inertial measurement unit (IMU), and GNSS. Integrating LiDAR with IMU demonstrates remarkable precision and robustness in high-dynamics and high-speed motions. However, LiDAR-Inertial systems encounter limitations in feature-scarce environments or during large-scale movements. GNSS integration overcomes these challenges by providing global and absolute measurements. LIGO employs an innovative hierarchical fusion approach with both front-end and back-end components to achieve synergistic performance. The front-end of LIGO utilizes a tightly coupled, extended Kalman filter (EKF)-based LiDAR-Inertial system for high-bandwidth localization and real-time mapping within a local-world frame. The back-end tightly integrates the filtered LiDAR-Inertial factors from the front-end with GNSS observations in an extensive factor graph, being more robust to outliers and noises in GNSS observations and producing optimized globally referenced state estimates. These optimized back-end results are then fed back to the front-end through the EKF to ensure a drift-free trajectory, particularly in degenerate and large-scale scenarios. Real-world experiments validate the effectiveness of LIGO, especially when applied to aerial vehicles with outlier-prone GNSS data, demonstrating its resilience to signal losses and data quality fluctuations. LIGO outperforms comparable systems, offering enhanced accuracy and reliability across varying conditions.

Index Terms—Global navigation, sensor fusion, simultaneous localization and mapping.

NOMENCLATURE

Back-End

\mathcal{X}_k System states optimized in the factor graph.
 $\mathbf{x}_n^f, n \in [0, k]$; States at various time steps included in \mathcal{X}_k

Received 17 September 2024; revised 6 December 2024; accepted 20 December 2024. Date of publication 15 January 2025; date of current version 4 February 2025. This work was supported by the Postgraduate Scholarships (PGS) from the University of Hong Kong. This article was recommended for publication by Associate Editor A. Nuechter and Editor J. Civera upon evaluation of the reviewers' comments. (*Corresponding author: Dongjiao He.*)

Dongjiao He and Haotian Li are with the Department of Mechanical Engineering, University of Hong Kong, Hong Kong, SAR 999077, China (e-mail: hdj65822@connect.hku.hk; haotianl@connect.hku.hk).

Jie Yin is with the Department of Electronic Engineering, Shanghai Jiao Tong University, Shanghai 200240, China (e-mail: 594666@sjtu.edu.cn).

Code for LIGO available online at <https://github.com/Joanna-HE/LIGO>.
Digital Object Identifier 10.1109/TRO.2025.3530298

Front-End

\mathbf{x}_k^l States optimized in the EKF.

I. INTRODUCTION

Simultaneous localization and mapping (SLAM) systems play a crucial role in enabling contemporary robots to undertake various autonomous tasks. Light detection and ranging (LiDAR) technology has seen widespread adoption in SLAM systems [1], [2] in recent years, owing to its exceptional performance in terms of high sampling rate (100 k to 1000 k Hz), accurate distance measurement (level of several millimeters to centimeters), and dense mapping capability. Extensive research has explored the potential of LiDAR sensors [3], accelerating the advancement of autonomous systems. LiDAR-based methods typically rely on the geometric structures captured by LiDAR points for tracking and constructing point maps.

Tracking methods based on geometric information can generally be classified into two categories. First are feature-based methods, which involve extracting features from the current scan (or frame) and then matching these features with consecutive scans, local maps, or global maps [1], [4], [5], [6], [7]. Second are direct methods, which register the current LiDAR points to features in the (local or global) map without extracting features for the current scan [8], [9], [10]. While feature-based approaches have shown impressive performance in structured urban and indoor scenes, often surpassing many vision-based methods in terms of localization accuracy [7], their performance is sensitive to environmental structures and relies heavily on accurate feature extraction in the current scan. In addition, these methods struggle to utilize previously measured LiDAR points that are sparsely distributed. On the other hand, direct methods, which directly register raw LiDAR points to the map, better adapt to structure-less environments. These methods can effectively utilize historical information in the global map for more accurate estimation. However, direct methods are typically sensitive to initial estimation.

Incorporating inertial measurement units (IMUs) into LiDAR-based odometry enhances robustness and efficiency while facilitating broader applications of direct methods by providing reliable initial estimations [2], [11]. Despite the use of direct matching methods in these systems, the output frequency remains constrained by the low frequency of LiDAR scans (e.g.,

10 Hz). Point-LiDAR inertial odometry (LIO) [12] employs a similar direct point-to-plane registration approach but achieves an exceptionally high output frequency ranging from 4 k to 8 k Hz by updating the state estimation at each registered LiDAR point. Furthermore, Point-LIO incorporates a stochastic-process augmented kinematic system, estimating acceleration and angular velocity as system states, which enhances robustness against noisy and saturated IMU measurements, enabling effective operation in high-dynamic motion scenarios.

Although robustness to incorrect LiDAR feature matches can be enhanced through proper fusion with IMU data, LiDAR-Inertial systems continue to experience drifts due to reliance on local measurements alone. Moreover, they are susceptible to degradation in environments with sparse structures or limited laser point coverage, particularly in outdoor settings, which restricts their effectiveness in certain scenarios. Previous studies have suggested addressing drifts through loop closure detection [13], [14] and pose graph optimization (PGO) [15], [16]. However, these methods are computationally expensive. For instance, as summarized in [14], loop closure description and its registration take between 46.5 and 376 ms for processing at a frequency of 1 Hz using a single thread on a system with an Intel i7-11700 k @ 3.6 GHz and 64 GB of memory. Similarly, Zou et al. [16] reported PGO times ranging from 32.8 to 83.3 ms for optimizing once after the loop closure detection on a system with an Intel i7-10700k @ 2.9 GHz and 15.5 GB of memory. Therefore, they are typically applied offline to generate a globally consistent map.

Alternatively, the global navigation satellite system (GNSS), offering absolute and global observations, can enhance localization performance in large-scale and less-structured environments. Nonetheless, GNSS suffers from low measurement frequency (1–10 Hz), significant outliers (tens of meters), and challenging noise modeling, presenting both complementary advantages and drawbacks compared to LiDARs. Integrating GNSS with LiDAR can concurrently address the accumulated drift and degradation issues encountered by LiDAR-Inertial systems in outdoor settings, and enable globally referenced estimation and consistent mapping leveraging the global and available observations from GNSS. Meanwhile, LiDAR-Inertial fusion facilitates localization and mapping in GNSS-denied environments such as indoor areas and tunnels. Moreover, the considerable outliers in GNSS data resulting from multipath effects and nonline-of-sight signals [17] can be mitigated by incorporating local constraints from the comparatively more precise LiDAR measurements. Simultaneously, the high-rate sampled LiDAR points contribute to increasing the system's output frequency.

Despite the promising outcomes of employing GNSS solutions directly, investigations into fusions of GNSS at the observation level are scarce [18], [19], [20], [21], [22]. While existing research on fusing GNSS observations with LiDAR-Inertial measurements has demonstrated some accuracy, achieving proper fusion of LiDAR and GNSS for their complementary effects remains a significant challenge. The five primary challenges persist, as follows.

- 1) Current systems typically fuse LiDAR using scan-to-scan or scan-to-submap feature registration, which relies on the sparse geometric structure of the current scan. This approach may lose constraints if features are sparse in the current scan.
- 2) Filter-based systems necessitate meticulous quality checks of GNSS measurements and are sensitive to outliers. Conversely, existing graph-based systems generate LiDAR registration factors at the same or lower rate as GNSS factors, resulting in odometry output at the low frequency of GNSS (e.g., 1 Hz, 5 Hz, 10 Hz). Such frameworks constrain odometry output frequency and are unsuitable for real-time odometry in GNSS-denied scenarios.
- 3) The robustness of current systems to loss of GNSS signals and LiDAR degeneracy remains largely unverified, leaving dynamic transitions among different scenarios untested.
- 4) Estimation accuracy, particularly for large-scale and high-speed motions prone to drifts of tens of meters in LiDAR-Inertial systems, could be further improved.
- 5) While experiments have been conducted primarily on vehicles and grounded robots, uncrewed aerial vehicles (UAVs) also require localization through GNSS assistance.

To address these challenges and advance the fusion of LiDAR, IMU, and GNSS, this article introduces LiDAR-inertial-GNSS odometry (LIGO), a tightly coupled LIGO system utilizing a hierarchical framework. The hierarchical framework consists of a front-end and a back-end designed to leverage the complementary advantages of LiDAR and GNSS. The key contributions of this work are outlined in the following five main aspects.

- 1) We introduce a novel tightly coupled LiDAR-Inertial-GNSS system, named LIGO, which achieves the following.
 - a) Competitive accuracy in trajectory estimation across large-scale scenarios with real-time mapping.
 - b) Robustness to either loss of GNSS signals or LiDAR degeneracy, enabling seamless handling of added or lost sensor signals during operation.
 - c) Capability of providing globally referenced pose estimations in both indoor and outdoor environments, suitable for ground vehicles and UAVs.
 - d) Robustness to noises and outliers in GNSS observations.
 - e) High-output-frequency odometry.
- 2) We propose a hierarchical framework to better exploit the complementary advantages of LiDAR and GNSS. The front-end utilizes an extended Kalman filter (EKF)-based tight fusion of LiDAR and IMU data in a local-world frame, providing real-time and high-frequency fusion results, including real-time mapping. The EKF is particularly effective in managing high-frequency and low-noise measurements, such as LiDAR points and IMU readings. In addition, the front-end provides values and covariance estimations referencing to the origin of the local-world

frame, facilitating the use of a compact LiDAR-Inertial factor, rather than independent LiDAR and inertial factors for fusing into a factor graph. The back-end employs a factor-graph-based tight fusion of GNSS observations (pseudoranges, Doppler shifts, and carrier phases) with LiDAR-Inertial results from the front-end. This factor graph utilizes locally accurate estimations (within centimeter-level precision) with their associated covariances from the front-end to tightly constrain GNSS observation outliers. The optimized state represented in the local-world frame is then fed back to the front-end to support drift-free trajectory, globally consistent mapping, and address the degradation of LiDAR measurement.

- 3) Our front-end is enhanced based on the implementation of Point-LIO [12] to address the increased measuring bandwidth for high-dynamic motions. Point-LIO directly registers current LiDAR points to the global map, removing the dependence on features of current scans and improving the robustness and accuracy in featureless scenarios. Moreover, using the point-by-point update scheme, odometry can be output at rates between 4 k and 8 k Hz. Furthermore, the augmented stochastic-process-based state formulation enables reliable performance under high-dynamic motion, where IMU measurements may be saturated or noisy, as seen in cases with angular velocities exceeding 75 rad/s.
- 4) Our back-end is lightweight yet effective, tightly fusing LiDAR-Inertial factors to mitigate outliers of GNSS observations in an extended factor graph, particularly in urban scenarios where multipath effects are common. In addition, the Hatch filter [23] is applied to exploit the more accurate carrier phase and improve the measurement precision of noisy and outlier-ridden pseudoranges. The single-difference factor of carrier phase used in [24] and [25] is also applied for its excellent performance.
- 5) Extensive experiments are conducted to validate the proposed method. These experiments encompass various indoor and outdoor scenarios, where LiDAR degrades or GNSS loses signals. Dynamic transitions among these scenarios are tested to further verify the robustness of our approach. Grounded and aerial datasets are collected and used for testing purposes.

The rest of this article is organized as follows. In Section II, we discuss relevant literature. Section III introduces the necessary information to realize the proposed system. The system overview of LIGO is presented in Section IV, and the detailed design of hierarchy formulation is illustrated in Section V. Subsequently, Section VI showcases the experiment results. Finally, Section VII concludes this article.

II. RELATED WORKS

As indicated in previous research [3], [26], both LiDAR-Inertial and visual-inertial systems are characterized by “locality,” providing precise and detailed information within local environments via lasers or images but lacking global information and susceptible to accumulated drifts in large-scale scenarios. In complement, GNSS offers absolute measurements in the

global Earth frame, suitable for incorporation with these local measurement systems to enhance performance.

Two popular methods for GNSS fusion are loosely coupled fusion and tightly coupled fusion. Loosely coupled fusion involves integrating GNSS solutions (i.e., position and velocity in Earth-north-up coordinates) with other sensor measurements, whereas tightly coupled fusion directly fuses GNSS raw observations such as pseudorange data, Doppler shifts, and carrier phases. Numerous efforts in loosely coupled frameworks can be found in the literature. For instance, the authors in [27], [28], [29], and [30] proposed generic multifusion systems based on the EKF, where the GNSS solution is fused with vision, LiDAR, inertial units, and other sensors such as wheel encoders. Similarly, Gao et al. [31] proposed a fast GNSS-LiDAR-Inertial state estimator, also based on the iterated EKF, which uses GNSS-derived solutions for coarse updates and LiDAR registration for finer estimation. This approach demonstrates a reduction in both absolute and relative positioning errors compared to several LiDAR-Inertial systems and a loosely coupled LiDAR-Inertial-global positioning system (GPS) system [32]. However, it depends on the accuracy of GNSS solutions and aligns estimated trajectories with ground truth via iterative closest point (ICP) before evaluating position accuracy, which limits its analysis of globally referenced state estimations. Shen et al. [33] presented a multisensor fusion framework extensible to different heterogeneous sensors, utilizing the unscented Kalman filter (UKF) where the absolute pose from the GPS is transformed to the local frame for feeding into the UKF as system outputs, leaving globally referenced state estimations unaddressed. GNSS solutions are loosely fused with local visual-inertial odometry or LiDAR-Inertial odometry in works such as [32], [34], [35], [36], [37] through graph optimization. LIO-Fusion [38] similarly employs a factor graph to integrate GNSS positions with LiDAR-Inertial-encoder odometry, showing estimations for globally referenced positions and improved performance in challenging environments like tunnels where LiDAR degradation occurs. Nevertheless, it relies on wheel encoders and prior LiDAR maps. In [39], a dual-graph framework is proposed to asynchronously fuse real-time kinematic (RTK) solutions with LiDAR and IMU measurements for construction tasks using excavators. This system addresses RTK dropout by switching between two distinct graphs, and it provides state outputs at the IMU sampling rate by propagating the state upon IMU data arrival. However, the noise from low-cost IMU sensors can degrade the accuracy of the propagated states. While loosely coupled fusion methods offer flexibility in implementation and computational efficiency, they suffer from poor accuracy performance for global reference estimation and sensitivity to outliers, such as corrupted GNSS solutions obtained with limited sky visibility.

Recent research studies have explored the concept of integrating individual GNSS satellite observations within a unified estimation framework instead of relying on precomputed GNSS fixes, aiming to enhance accuracy and robustness. Among the most utilized GNSS observations are pseudorange data, Doppler shifts, and carrier phases. Pseudorange data represent range-only observations between satellites and GNSS receivers, often used for absolute trajectory anchors in a global Earth frame and to

mitigate odometry drifts, despite typically exhibiting larger uncertainties on the order of meters. Doppler shift measurements, characterized by lower outliers and noises, and carrier phases, with outliers at the centimeter scale and noises at millimeter scale, are used to supplement pseudorange data for improved accuracy, especially in achieving local smoothness. Notably, Liu et al. [40] presented the first attempt to tightly fuse pseudorange and Doppler shifts with a visual-inertial system, employing an optimization-based approach. GNSS visual inertial navigation system (GVINS), introduced in [41], tightly fuses pseudorange measurements and Doppler shifts with the previous visual-inertial system, Vins-mono [42]. Similarly, the authors in [43] and [44] achieved this fusion using an EKF-based method, considering observability constraints. In addition, the authors in [45] and [24] incorporated carrier phase measurements into the tight fusion framework.

Despite numerous works focusing on the tight coupling of GNSS observations with visual-inertial systems, similar efforts for LiDAR are scarce [18], [19], [20], [21], [22]. Li et al. [18] achieved tight fusion of GNSS observations (pseudorange and carrier phase) with IMU and LiDAR using a multistate constraint Kalman filter. However, this filter-based tight fusion is susceptible to GNSS outliers, making it challenging to maintain continuously precise estimations, particularly in urban canyons. Li et al. [19] employed the EKF to fuse pseudorange, carrier phase, IMU, and LiDAR, utilizing a sliding-window plane-feature tracking method to enhance efficiency, which also suffers from the performance degradation caused by the GNSS outliers. Li et al. [20] employed a factor graph to tightly fuse LiDAR, IMU, and GNSS data, achieving tens-of-centimeter-level accuracy on self-collected datasets, albeit relying on base stations to provide differential constraints. Similarly, Liu et al. [22] tightly fuse pseudorange measurements and Doppler shifts with LiDAR and IMU in GLIO, relying on pseudorange measurements from base stations. GLIO fuses a double-difference pseudorange factor between the receiver and the station to estimate the poses in the local frame, necessitating a preknown anchor pose for globally referenced localization. Beuchert et al. [21] proposed a factor-graph-based tight fusion of LiDAR, IMU measurements, and GNSS observations without dependence on base stations or preknown anchor poses. By employing a double-difference carrier phase factor previously formulated in [46], the proposed method eliminates the explicit initialization stage and achieves meter-level accuracy even in environments with poor GNSS visibility. However, the system's output frequency remains constrained by the GNSS sampling rate, and its performance has not been validated in challenging conditions such as indoor environments with GNSS signal loss or scenarios with LiDAR degradation.

To address the limitations of existing systems, we propose a tightly coupled LiDAR-Inertial-GNSS system that operates independently of base stations or preknown information, while being comparably more robust to GNSS observation outliers for state estimation and capable of handling high measuring bandwidth for highly dynamic motions. In contrast to [21], our focus lies on introducing an innovative tight fusion framework to leverage the complementary properties of both LiDAR and

GNSS systems. As illustrated in Table I, significant differences exist between the characteristics of LiDAR and GNSS systems. On the one hand, LiDAR offers extremely high-frequency odometry and mapping output of thousands Hz due to its high sampling rate (i.e., 100 k to 1000 k Hz). In addition, its accurate depth measurements can enhance performance in high-dynamic and high-speed motions. However, directly fusing LiDAR factors into a factor graph might suppress these potentials. On the other hand, GNSS observations can improve estimation accuracy and provide globally referenced localization through direct measurements in the Earth-central frame, supporting LiDAR-based systems, particularly in outdoor environments. Nonetheless, GNSS observations are prone to high levels of outliers and noises. Furthermore, when fusing GNSS with local measurements, the extrinsic of local motions within the global frame of GNSS could be orders of magnitude larger than the scale of the local movements. Given these characteristics of the GNSS system, a filter-based framework would be ineffective. To address the distinct characteristics of LiDAR and GNSS systems, we propose a novel hierarchical tight fusion framework. We conduct extensive experiments to validate the performance of the proposed LiDAR-Inertial-GNSS system, including its ability to handle high outlier-laden and noisy GNSS observations, seamless operation under addition/loss of sensor signals, and its applicability to UAVs. Notably, none of these functionalities have been verified in previous works.

III. PRELIMINARY

A. Coordinate Systems

The multisensor fusion system proposed in this article requires five coordinates to fuse measurements from GNSS, LiDAR, and IMU sensors. We denote $^E(\cdot)$, $^w(\cdot)$, $^l(\cdot)$, $^i(\cdot)$, and $^r(\cdot)$ as the 3D variables in the Earth-centered, Earth-fixed (ECEF) coordinate; in the local-world coordinate; in the LiDAR coordinate; in the IMU coordinate; and in the GNSS receiver coordinate, respectively. ECEF is a right-handed Cartesian frame fixed to the Earth, with its z -axis pointing to the geographical North Pole and its x -axis pointing to the prime meridian. The local-world frame is also fixed to the Earth by coarsely aligning the z -axis of the initial IMU frame with gravity at the beginning stage. Furthermore, the LiDAR, IMU, and GNSS receiver frames are all attached to their respective sensors. An illustration of these frames is provided in Fig. 1.

B. Point-LIO

Point-LIO [12] is a LiDAR-Inertial odometry system that tightly integrates LiDAR and IMU data within an EKF. Notably, Point-LIO boasts a high bandwidth tailored for high-dynamic and high-speed motions. It adopts a point-by-point framework that updates system states at each LiDAR point, directly registering it to the global map. This approach leverages the high sampling rate of LiDAR data, thereby achieving a high odometry and mapping output frequency ranging from 4 k to 8 k Hz, and effectively eliminating motion distortion. Point-LIO also

TABLE I
COMPARISON OF THE CHARACTERS OF LiDAR AND GNSS

Characters	Outliers	Noises	Sampling Rate	Local/Global	Ability of Mapping	Need of Environment Structures
LiDAR	Few	Small	High	Local	Yes	Yes
GNSS	Many	Large	Low	Global	No	No

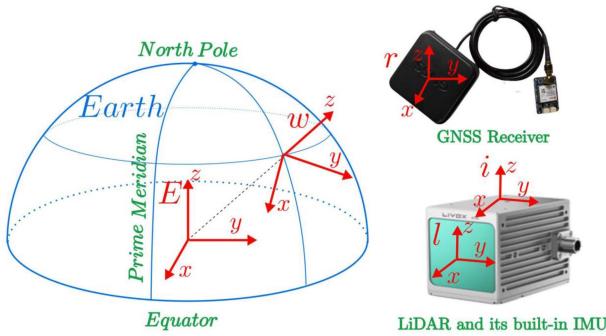


Fig. 1. Illustration of the coordinates. E , w , r , l , and i denote the ECEF, local-world, GNSS receiver, LiDAR, and the IMU coordinate, respectively.

uses a stochastic process augmented model, allowing it to harness the precise measurements from LiDAR to compensate for saturated and noisy IMU readings during aggressive motions that surpass the IMU's measuring limits. In this article, we implement the LIGO system by incorporating the principles of Point-LIO, thereby preserving the high bandwidth characteristic of Point-LIO for high-dynamic and high-speed motions.

C. Hatch Filter

In general, the distance between satellites and GNSS receivers can be determined through two primary methods: pseudorange measurements and carrier phase measurements. While the accuracy of pseudorange measurements is relatively lower compared to carrier phase measurements, pseudorange measurements allow for absolute distance measurement in the Earth-central frame. On the other hand, carrier phase measurements offer higher accuracy but involve an unknown integer ambiguity, precluding absolute distance detection. Therefore, combining both methods presents a natural approach to achieve more accurate and absolute distance measurements. One of the widely used integration techniques is the carrier-smoothed code (CSC) technique [47]. As a result, various CSC algorithms have been developed for mobile GNSS applications. Among these, the Hatch filter [48] stands out as one of the most renowned and simplest filtering approaches.

A single-frequency Hatch filter simultaneously utilizes pseudorange measurements and carrier phase measurements. Its operation within a sample window of size N can be expressed as follows:

$$\mathcal{H}(k) = \frac{1}{n} \mathcal{P}(k) + \frac{n-1}{n} (\mathcal{H}(k-1) + \phi(k) - \phi(k-1)), \quad k \geq 1$$

$$\mathcal{H}(0) = \mathcal{P}(0).$$

Here, \mathcal{P} denotes the pseudorange measurement, ϕ denotes the carrier phase measurement, and \mathcal{H} represents the Hatch filter measurement, which is a carrier-smoothed pseudorange measurement. The parameter k denotes the time step of the measurements from the initial measurement without any cycle slip or signal loss. Specifically, $n = k$ when $k < N$ and $n = N$ when $k \geq N$.

Following the filtering process, the noise in the Hatch filter measurement $\sigma_{\mathcal{H}}$ can be characterized by the pseudorange measurement noise $\sigma_{\mathcal{P}}$ and the carrier phase measurement noise σ_{ϕ} as follows:

$$\sigma_{\mathcal{H}} = \sqrt{\frac{1}{2n} \sigma_{\mathcal{P}}^2 + \sigma_{\phi}^2}. \quad (1)$$

Given a sufficiently large time constant, the noise in the carrier-smoothed pseudorange measurement approaches that of the carrier phase measurement.

IV. SYSTEM OVERVIEW

A. Notation

In this article, we use ${}^a_b(\cdot)$ to denote the value of variable (\cdot) of the frame b represented in the frame a , and ${}^a(\cdot)$ to denote the value of variable (\cdot) represented in the frame a . We use the subscript $(\cdot)_a$ to refer to the variable (\cdot) at time step a .

For the state representation, the notations used in this article are summarized in the Nomenclature.

B. States

We denote the system state at time step k by \mathcal{X}_k , expressed as

$$\mathcal{X}_k = \left[\mathbf{x}_n^f \quad {}^E_w \mathbf{T}_k \right], \quad n \in \mathcal{T}_k. \quad (2)$$

Here, ${}^E_w \mathbf{T}_k = ({}^E_w \mathbf{R}_k, {}^E_w \mathbf{p}_k)$ represents the extrinsic state of the local-world frame to the ECEF frame at time step k . The \mathcal{T}_k denotes the time window of states estimated at time step k , and index n signifies the time step. Specifically

$$\mathbf{x}_n^f = \left[{}^w_i \mathbf{R}_n \quad {}^w_i \mathbf{p}_n \quad {}^w_i \mathbf{v}_n \quad {}^w_g \mathbf{g}_n \quad {}^i \mathbf{b}_{g_n} \quad {}^i \mathbf{b}_{a_n} \quad {}^i \omega_n \quad {}^i \mathbf{a}_n \quad \delta \mathbf{t}_n \quad \dot{\delta \mathbf{t}}_n \right]^T.$$

In the abovementioned equation, ${}^w_i \mathbf{R}_n \in \text{SO}(3)$, ${}^w_i \mathbf{p}_n \in \mathbb{R}^3$, and ${}^w_i \mathbf{v}_n \in \mathbb{R}^3$ represent the position, rotation, and velocity of the IMU frame i with respect to the local-world frame w at time step n . In addition, ${}^i \mathbf{a}_n \in \mathbb{R}^3$ and ${}^i \omega_n \in \mathbb{R}^3$ denote the acceleration and angular velocity in the IMU frame, while ${}^i \mathbf{b}_{a_n} \in \mathbb{R}^3$ and ${}^i \mathbf{b}_{g_n} \in \mathbb{R}^3$ represent the IMU biases. Gaussian noise drives all four states. Furthermore, ${}^w_g \mathbf{g}_n \in \mathbb{R}^3$ signifies gravity in the local-world frame. $\delta \mathbf{t}_n \in \mathbb{R}^4 = [\delta t_{g_n} \quad \delta t_{b_n} \quad \delta t_{a_n} \quad \delta t_{o_n}]^T$ is a vector containing the clock offsets of the receiver clock.

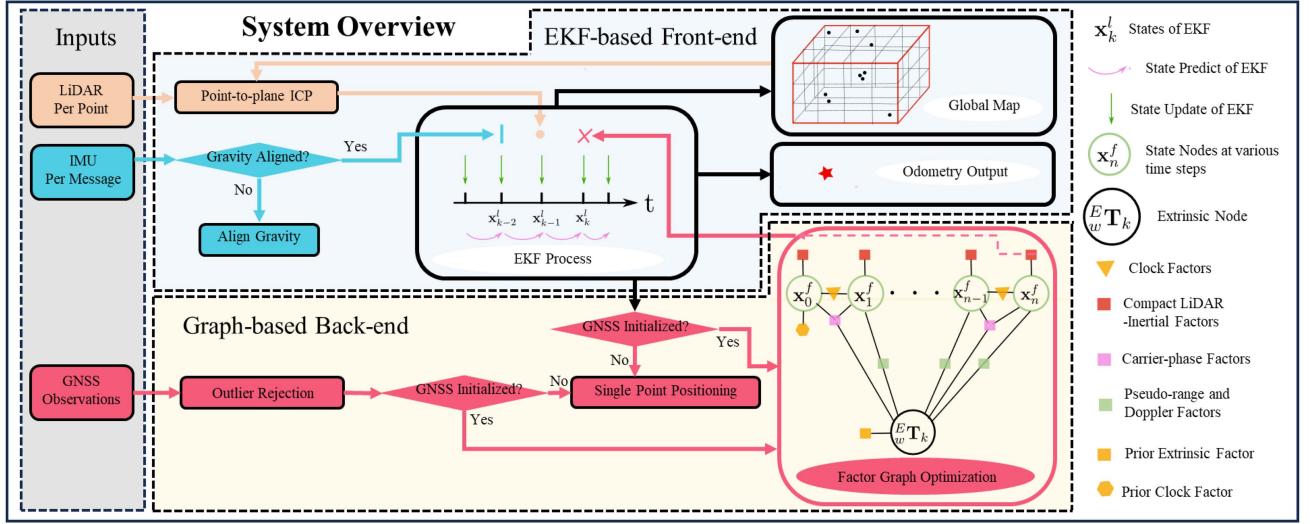


Fig. 2. System overview of LIGO, featuring a hierarchical formulation with a front-end and a back-end. The front-end, based on the EKF, tightly fuses LiDAR and IMU measurements. Initially, IMU measurements accumulate to estimate the initial gravity value, aligning the z -axis of the initial IMU frame with gravity to establish the local-world frame. Subsequently, the front-end employs a point-by-point state update mechanism in the local-world frame. The front-end also maintains a global map by projecting LiDAR points using estimated poses. The state with distribution estimation results of the front-end provides a compact LiDAR-Inertial factor to be fused in back-end. The back-end processes outlier rejection for GNSS observations. It initializes translation extrinsic and clock drifts using the SPP algorithm. Postinitialization, states are optimized in the factor graph by fusing compact LiDAR-Inertial factors from the front-end and GNSS factors. Finally, optimized values from the back-end, update the EKF states and also the global map.

Four independent and fully operational GNSS systems are considered: GPS with δt_{g_n} , BeiDou satellite navigation system (BDS) with δt_{b_n} , GALILEO satellite navigation system (GALILEO) with δt_{a_n} , and GLONASS satellite navigation system (GLONASS) with δt_{o_n} . δt_n represents the receiver clock drifting rate.

For the following sections, we further denote \mathbf{x}_k^l as the state, identified by “locality,” with partial components of the state \mathbf{x}_n^f , where

$$\mathbf{x}_k^l = \begin{bmatrix} {}_i^w\mathbf{R}_k & {}_i^w\mathbf{p}_k & {}_i^w\mathbf{v}_k & {}^w\mathbf{g}_k & {}^i\mathbf{b}_{g_k} & {}^i\mathbf{b}_{a_k} & {}^i\mathbf{a}_k & {}^i\omega_k \end{bmatrix}. \quad (3)$$

C. Initialization

Although a carefully designed initialization model can contribute to accurate initial state estimation and the subsequent successful operation of the state estimation system, such a model necessitates specific and adequate motion excitations, as well as a model switch during system operation [41]. Moreover, this initialization step may not be essential for certain systems, as the optimization solution of the system itself can achieve state convergence.

Therefore, LIGO employs a coarse two-step initialization model to facilitate online application. At the first step, ${}^w\mathbf{g}$ is roughly initialized in the local-world frame by computing the negative value of the average acceleration measurements obtained from the IMU. Subsequently, the local-world frame is established by aligning the z -axis of the initial IMU frame with the direction of gravity. Then, at the second step, the translation extrinsic of the local-world frame in the ECEF frame, along with the time clock drifts, are determined from the pseudorange

measurements using the single point positioning (SPP) algorithm (see the supplementary file [49] for an illustration.), the same SPP utilized in GVINS [41].

D. System Design Motivation and Overview

Sensor fusion can be achieved using filtering or factor graph techniques, each suited to different measurement types. Filtering methods, based on a Markov process, are efficient but sensitive to outliers and noise. Factor graphs, by considering factors over time, excel in noise mitigation and outlier rejection. However, to remain efficient, factor graphs must reduce the number of estimated states, potentially limiting system output frequency [50].

To combine the complementary strengths of LiDAR and GNSS, as shown in Table I, we propose a hierarchical approach with an EKF-based front-end and a graph-based back-end to fuse LiDAR, IMU, and GNSS measurements. This results in a tightly coupled LIGO system, where the front-end integrates LiDAR and IMU data tightly, and the back-end further fuses GNSS observations with LiDAR and IMU tightly.

Fig. 2 depicts the system overview of LIGO. As described in Section IV-C, LIGO uses a coarse two-step initialization to launch the system. After the first step, the front-end operates as an EKF-based, tightly coupled LiDAR-Inertial SLAM system, with the system state defined in (3). Upon completion of the second step, the front-end’s fused states and their EKF-estimated covariances are compacted into a LiDAR-Inertial factor and tightly integrated with GNSS observations within the back-end factor graph, with system state as depicted in (2). Finally, the back-end’s optimized results are fed back to the front-end as

system outputs and the map structure is modified according to the state modifications.

For LiDAR measurements, given their high accuracy and sampling rate, they are tightly integrated with IMU data in the EKF-based front-end (see Section V-A), to deliver high-frequency odometry and mapping outputs with a large measuring bandwidth suitable for dynamic motions. After initializing gravity and establishing the local-world frame, leveraging information from IMU and LiDAR measurements, the front-end optimizes system states \mathbf{x}^l as defined in (3) following the state predict and state update filtering traditions. The front-end, using the state estimator of Point-LIO [12] as elaborated in Section III-B, ensures high bandwidth for high-dynamic motions. The map of the surroundings is constructed with LiDAR points projected by estimated poses in the local-world frame. In addition, state and covariance estimations relative to the local-world frame are compacted into LiDAR-Inertial factors for seamless integration in the back-end.

For GNSS observations, which provide global positioning but are affected by outliers and low sampling rates, they are fused with the front-end results using a factor graph to enhance information extraction, mitigate outliers, and ensure stable estimation of extrinsics between the local-world and global frames. In the back-end, GNSS observations, including pseudorange, Doppler shifts, and carrier phase, are processed with outlier rejection. The back-end (see Section V-B) initializes translation extrinsic and GNSS receiver clock drifts using the SPP algorithm. Then, it fuses compact LiDAR-Inertial factors and GNSS factors at the GNSS observation frequency, optimizing all states \mathcal{X}_k as defined in (2) within a fixed-size time window \mathcal{T}_k , to estimate extrinsic parameters and achieve global-referenced localization.

Finally, the latest optimized results from the back-end, capturing GNSS information, are fed back to the front-end as EKF outputs to assist in consistent state and map modifications. Upon each update, poses within a time interval from the last GNSS observation to the current are fitted and optimized in a B-spline curve. The local structure of the global map is then updated by querying B-spline poses and reprojecting LiDAR points. This ensures drift-free trajectories and robust operation even under LiDAR degeneracy.

It is important to note that the odometry and mapping results are generated by the front-end to maintain a high output frequency. These front-end results are not the final outputs of LIGO, as the back-end continues to optimize the states within its time window. However, the accuracy of the front-end outputs is sufficient for most real-time applications.

V. HIERARCHY FORMULATION OF PROBLEM

This section outlines the hierarchical framework of LIGO, encompassing both the front-end and back-end components.

A. EKF-Based Front-End

Point-LIO [12] with modified map structure serves as the front-end in this system, employing an EKF-based LiDAR-Inertial approach.

1) *State Estimation*: The state estimated in the front-end is \mathbf{x}^l as defined in (3) in Section IV-B. A specialized EKF, designed to evolve on general manifolds [51], is employed to fuse both IMU measurements and LiDAR points, consistent with the approach outlined in Section III-B. Notably, the EKF estimates the system state with an associated distribution, enabling the tight integration of these estimates with measurements from other sensors.

2) *Mapping*: We adopt the grid map methodology described in [52] as our global map, enabling efficient search and insert operations. This grid map divides the spatial area into uniform voxels and represents the environment using points. Leveraging the precise measurements from LiDAR sensors, we construct the map by projecting the measured LiDAR points using the current optimized poses into the local-world frame.

3) *Analysis*: The proposed front-end offers several notable advantages, as follows.

- 1) The EKF-based framework is well-suited for accurate and high-frequency LiDAR measurements. Extensive evaluations in formal LiDAR-based SLAM systems [2], [12], [53] have demonstrated that EKF-based methods deliver comparable or superior accuracy compared to factor-based methods, while significantly improving efficiency.
- 2) The front-end inherits the superior properties of Point-LIO, supporting real-time and high-frequency odometry and map outputs. It is robust in handling saturated and noisy IMU measurements during aggressive motions, and it excels in unstructured environments due to its direct registration of LiDAR points to the global map.
- 3) The EKF framework in the front-end generates state estimations with noise distributions by integrating LiDAR and IMU measurements. These estimations can be further fused with other measurements in a tight manner, forming a compact LiDAR-Inertial factor. This fusion capability extends to both EKF and factor graph frameworks.

B. Factor-Graph-Based Back-End

In the back-end, our approach involves the fusion of LiDAR, IMU, and GNSS measurements within a factor graph. Given the outlier-ridden, noisy, and low-frequency nature of GNSS measurements, as well as the significant scale differences between GNSS (tens of millions) and LiDAR measurements (hundreds or thousands), we propose integrating GNSS with LiDAR and IMU data within the factor graph. Leveraging the compact factor representation of LiDAR and IMU measurements obtained from the front-end, we extend the window size of the factor graph. This extension enables a more accurate fusion of GNSS measurements, ensuring better overall system performance.

1) *Problem Formulation*: With all measurements considered, the back-end seeks to maximize a posterior (MAP) of the entire state \mathcal{X}_k . Assuming the independence of each measurement and zero-mean Gaussian distribution for the noise associated with each measurement, the MAP problem can be reformulated to minimize the sum of a set of costs. Each cost

corresponds to a specific measurement

$$\begin{aligned}\hat{x}_k &= \arg \max_{x_k} \mathbf{p}(x_k | z_0, \dots, z_k) \\ &= \arg \max_{x_k} \mathbf{p}(x_k) \mathbf{p}(z_0, \dots, z_k | x_k) \\ &= \arg \max_{x_k} \mathbf{p}(x_k) \mathbf{p}(z_0 | x_k), \dots, \mathbf{p}(z_k | x_k) \\ &= \arg \min_{x_k} \left(\| \mathbf{r}_p - \mathbf{H}_p x_k \|_{\mathbf{P}_p}^2 + \sum_{n=0}^k \| \mathbf{r}(x_k, z_n) \|_{\mathbf{P}_n}^2 \right)\end{aligned}$$

where \mathbf{r}_p and \mathbf{H}_p encapsulate the prior information of the system state. $\mathbf{r}(x_k, z_n)$ denotes the residual function of n th independent measurement and $\| \cdot \|_{\mathbf{P}}^2$ is the Mahalanobis norm.

Fig. 2 shows the factor graph of the system, which connects states and measurements with an exact dependent relation. In addition, a prior factor is used to confine the initial states, encompassing the initial pose in the local-world frame, the initial receiver clock drifts and their drift rate, and the extrinsic parameters of the local-world frame in the ECEF frame. Subsequently, we will discuss each factor in detail.

2) *Compact LiDAR-Inertial Factor*: The compact LiDAR-Inertial factor originates from the estimation outcomes of the EKF-based front-end.

On the one hand, the estimations from the front-end, denoted as \hat{x}_k^l , are defined within the local-world frame, with their corresponding covariances reflecting the uncertainties of \hat{x}_k^l relative to the ground truth $x_k^{l,g}$ in the local-world frame, as well as the correlation among components of the state x_k^l . On the other hand, the states estimated in the back-end factor graph are divided into three components: ${}^E_w \mathbf{T}_k$, representing the extrinsics of the local-world frame with respect to the global frame; \mathbf{x}_k^f , representing the time-varying states, including two components, which are \mathbf{x}_k^l and the GNSS receiver related states (i.e., clock drifts and their drifting rate). Among these components, LiDAR and IMU measurements only observe the values and correlations of \mathbf{x}_k^l , without constraining the extrinsics or the receiver-related states. Moreover, the information from LiDAR and IMU measurements is fully incorporated into the state estimations in the local-world frame and their covariances, allowing for further correlation with other components. Consequently, the compact LiDAR-Inertial factor should be modeled as a “Prior Factor,” where the measurement value is \hat{x}_k^l , and the measurement noise is represented by the estimation covariance, thereby preserving the correlations among the components of \mathbf{x}_k^l .

Consequently, the measurement \mathbf{z}_k^e can be formulated as follows:

$$\mathbf{z}_k^e = \hat{x}_k^l = [{}^w \hat{\mathbf{R}}_k \ {}^w \hat{\mathbf{p}}_k \ {}^w \hat{\mathbf{v}}_k \ {}^w \hat{\mathbf{g}}_k \ {}^i \hat{\mathbf{b}}_{g_k} \ {}^i \hat{\mathbf{b}}_{a_k} \ {}^i \hat{\omega}_k \ {}^i \hat{\mathbf{a}}_k]^T$$

and the measurement covariance is the estimated covariance of \hat{x}_k^l from the front-end.

Finally, the residual connecting the system states and \mathbf{z}_k^e can be formulated as follows:

$$\begin{aligned}\mathbf{r}(x_k, \mathbf{z}_k^e) &= \mathbf{x}_k^l \boxminus \hat{x}_k^l \\ &= [\log({}^i \hat{\mathbf{R}}_k {}^i \mathbf{R}_k) \ {}^i \hat{\mathbf{p}}_k - {}^i \mathbf{p}_k \ {}^i \hat{\mathbf{v}}_k - {}^i \mathbf{v}_k]\end{aligned}$$

$$\begin{aligned}& {}^w \hat{\mathbf{g}}_k - {}^w \mathbf{g}_k \ {}^i \hat{\mathbf{b}}_{g_k} - {}^i \mathbf{b}_{g_k} \ {}^i \hat{\mathbf{b}}_{a_k} - {}^w \mathbf{b}_{a_k} \\ & {}^i \hat{\omega}_k - {}^i \omega_k \ {}^i \hat{\mathbf{a}}_k - {}^i \mathbf{a}_k]^T\end{aligned}$$

where $\log(\cdot)$ is the operation to calculate the axis angle of the rotation matrix, and \mathbf{x}_k^l is the variable in the factor graph.

Owing to the fusion outcomes of the EKF, the degradation of LiDAR can be directly discerned through the state covariance, obviating the necessity for a meticulously crafted model for degradation detection.¹ In addition, the tight integration of LiDAR with IMU ensures that the state covariance would not become infinite if degradation arises in one or more directions of LiDAR measurements. There are other superior properties of the compact LiDAR-Inertial factors, that is, the compact LiDAR-Inertial factor aids in reducing the number of factors in the graph by $\mathcal{O}(N)$, where N represents the number of variables optimized in the graph, and this factor also facilitates the provision of more robust LiDAR constraints in structure-less and repetitive environments.

3) *Single-Difference Carrier Phase Factor*: The carrier phase observation, represented as $\phi_k^{m,j}$, is transmitted by the j th satellite of the m th GNSS system at time step k . It constitutes a precise and direct distance measurement between the satellite and the receiver, with its expression given as follows:

$$\begin{aligned}\phi_k^{m,j} &= \| {}^E \mathbf{p}_k^j - {}^E \mathbf{p}_k \| + \mathcal{T}_k^j - \mathcal{I}_k^j + c(\delta t_k^m - \delta t_k^j - \nu_j^j) \\ &\quad + \lambda_j(\kappa_j - N_j)\end{aligned}\quad (4)$$

where $\| {}^E \mathbf{p}_k^j - {}^E \mathbf{p}_k \|$ represents the distance between the satellite and GNSS receiver, \mathcal{T} and \mathcal{I} denote the tropospheric and ionospheric delays in meters, respectively, which can be derived by decoding the satellite signals. c denotes the light speed in meters per second, and δt_k^m signifies the receiver clock drift of the m th GNSS system (e.g., GPS, BDS, GALILEO, or GLONASS). In addition, δt_k^j stands for the clock drift of satellite j , which is also transmitted via the GNSS signal. ν_j is the relativity term, defined as $\nu_j = -2 \frac{({}^E \mathbf{p}_k^j)^T {}^E \mathbf{v}_k^j}{c^2}$, with ${}^E \mathbf{v}_k^j$ as the satellite j 's velocity. κ_j arises from the interaction between the changing satellite orientation and the circularly polarized carrier wave, with wavelength λ_j , while N_j denotes the integrity ambiguity, an unknown offset.

Although the carrier phase observation boasts superior precision, the integrity ambiguity remains unknown, necessitating the determination of this value by a ground station to render the carrier phase observation meaningful. To alleviate dependency on the ground station, a single difference of two carrier signals obtained at different time steps is employed.

The two signals (at time step k and $l < k$) from the same satellite, locked in the same cycle, share the same ambiguity number N . Consequently, the integrity ambiguity can be eliminated by taking the difference between them, as illustrated in

¹In theory, smaller covariance indicates better constraint in that direction. LiDAR degradation implies a loss of constraints, leading to larger covariance along the degraded directions. While the EKF's inconsistency impacts the accuracy of the covariance estimation, the principle remains that smaller covariance reflects better-constrained states. The inconsistency of EKF varies by systems, and enhancing EKF consistency is beyond the scope of this article.

the following:

$$\begin{aligned}\phi_k^{m,j} - \phi_l^{m,j} &= \left(\|{}^E\mathbf{p}_k^j - {}_i^E\mathbf{p}_k\| + \mathcal{T}_k^j - \mathcal{J}_k^j \right. \\ &\quad \left. + c(\delta t_k^m - \delta t_k^j - \nu_k^j) + \lambda_j(\kappa_j - N_j) \right) \\ &\quad - \left(\|{}^E\mathbf{p}_l^j - {}_i^E\mathbf{p}_l\| + \mathcal{T}_l^j - \mathcal{J}_l^j \right. \\ &\quad \left. + c(\delta t_l^m - \delta t_l^j - \nu_l^j) + \lambda_j(\kappa_j - N_j) \right) \\ &= \left(\|{}^E\mathbf{p}_k^j - {}_i^E\mathbf{p}_k\| + \mathcal{T}_k^j - \mathcal{J}_k^j + c(\delta t_k^m - \delta t_k^j - \nu_k^j) \right) \\ &\quad - \left(\|{}^E\mathbf{p}_l^j - {}_i^E\mathbf{p}_l\| + \mathcal{T}_l^j - \mathcal{J}_l^j + c(\delta t_l^m - \delta t_l^j - \nu_l^j) \right).\end{aligned}$$

Furthermore, given that \mathcal{T} , \mathcal{J} , δt^j , and ν^j are observable in the GNSS signals, the measurement model of the single-difference carrier phase factor is formulated as follows:

$$\begin{aligned}\mathbf{z}_k^{cp} &= \phi_k^{m,j} - \phi_l^{m,j} - \mathcal{T}_k^j + \mathcal{J}_k^j + c(\delta t_k^j + \nu_k^j) + \mathcal{T}_l^j - \mathcal{J}_l^j - c(\delta t_l^j + \nu_l^j) \\ &= \|{}^E\mathbf{p}_k^j - {}_i^E\mathbf{p}_k\| + c\delta t_k^m - \|{}^E\mathbf{p}_l^j - {}_i^E\mathbf{p}_l\| + c\delta t_l^m\end{aligned}$$

and the residual is calculated as

$$\begin{aligned}\mathbf{r}(\mathcal{X}_k, \mathbf{z}_k^{cp}) &= \|{}^E\mathbf{p}_k^j - {}_w^E\mathbf{T}_k {}_i^w\mathbf{p}_k\| + c\delta t_k^m \\ &\quad - \|{}^E\mathbf{p}_l^j - {}_w^E\mathbf{T}_k {}_i^w\mathbf{p}_l\| - c\delta t_l^m - \mathbf{z}_k^{cp}\end{aligned}$$

where ${}_w^E\mathbf{T}_k = ({}_w^E\mathbf{R}_k, {}_w^E\mathbf{p}_k)$ is the extrinsic between the local-world frame and the ECEF frame at time step k .

4) CSC Factor: The CSC factor utilizes the Hatch filter. For a signal from satellite j of a GNSS system m at time step k , its measurement is represented as $\mathcal{H}_k^{m,j}$, with the formulation detailed in Section III-C

$$\begin{aligned}\mathcal{H}_k^{m,j} &= \frac{1}{n} \mathcal{P}_k^{m,j} + \frac{n-1}{n} \left(\mathcal{H}_{k-1}^{m,j} + \phi_k^{m,j} - \phi_{k-1}^{m,j} \right), k \geq 1, \\ \mathcal{H}_0^{m,j} &= \mathcal{P}_0^{m,j}\end{aligned}\quad (5)$$

where $\mathcal{P}_k^{m,j}$ denotes the pseudorange measurements at time step k , with its expression elaborated as

$$\mathcal{P}_k^{m,j} = \|{}^E\mathbf{p}_k^j - {}_i^E\mathbf{p}_k\| + \mathcal{T}_k^j + \mathcal{J}_k^j + c(\delta t_k^m - \delta t_k^j - \nu_k^j). \quad (6)$$

The expression of the pseudorange measurement resembles that of the carrier phase, with the distinction that pseudorange measurements offer absolute distance measurements between the satellite and the receiver without the presence of unknown integrity ambiguity. Nonetheless, pseudorange measurements are prone to significant noises and outliers.

By substituting (6) and (4) into (5), we obtain $\mathcal{H}_k^{m,j}$ with an expression identical to that of the pseudorange measurements, albeit with mitigated noises

$$\mathcal{H}_k^{m,j} = \|{}^E\mathbf{p}_k^j - {}_i^E\mathbf{p}_k\| + \mathcal{T}_k^j + \mathcal{J}_k^j + c(\delta t_k^m - \delta t_k^j - \nu_k^j).$$

Given that \mathcal{T} , \mathcal{J} , δt_k^j , and ν_k^j are determined through the decoding of GNSS signals, the measurement model of the CSC factor can be reformulated as

$$\begin{aligned}\mathbf{z}_k^h &= \mathcal{H}_k^{m,j} - \mathcal{T}_k^j - \mathcal{J}_k^j + c(\delta t_k^j + \nu_k^j) \\ &= \|{}^E\mathbf{p}_k^j - {}_i^E\mathbf{p}_k\| + c\delta t_k^m.\end{aligned}$$

Finally, the CSC factor is related to the states as

$$\mathbf{r}(\mathcal{X}_k, \mathbf{z}_k^h) = \|{}^E\mathbf{p}_k^j - {}_w^E\mathbf{T}_k {}_i^w\mathbf{p}_k\| + c\delta t_k^m - \mathbf{z}_k^h. \quad (7)$$

5) Doppler Factor: The Doppler frequency shifts measure the relative velocity of the satellite and receiver along the line of signal propagation path, as depicted by

$$\begin{aligned}\mathcal{D}_k^j &= -\frac{1}{\lambda_j} \left(\frac{{}^E\mathbf{p}_k^j - {}_i^E\mathbf{p}_k}{\|{}^E\mathbf{p}_k^j - {}_i^E\mathbf{p}_k\|} \right)^T \left({}^E\mathbf{v}_k^j - {}_i^E\mathbf{v}_k \right) \\ &\quad - \frac{c}{\lambda_j} (\dot{\delta t}_k - \dot{\delta t}_k^j)\end{aligned}$$

where $\dot{\delta t}_k^j$ represents the clock drift rate of the j th satellite at time step k , λ_j denotes the wavelength, and ${}^E\mathbf{v}_k^j$ signifies the velocity of satellite j . Given that λ_j and i_k^j are known, the measurement model of the Doppler frequency shifts can be reconfigured as

$$\begin{aligned}\mathbf{z}_k^d &= \lambda_j \mathcal{D}_k^j - c\dot{\delta t}_k^j = -\left(\frac{{}^E\mathbf{p}_k^j - {}_i^E\mathbf{p}_k}{\|{}^E\mathbf{p}_k^j - {}_i^E\mathbf{p}_k\|} \right)^T \\ &\quad \times \left({}^E\mathbf{v}_k^j - {}_i^E\mathbf{v}_k \right) - c\dot{\delta t}_k\end{aligned}$$

and the residual is

$$\begin{aligned}\mathbf{r}(\mathcal{X}_k, \mathbf{z}_k^d) &= -\left(\frac{{}^E\mathbf{p}_k^j - {}_w^E\mathbf{T}_k {}_i^w\mathbf{p}_k}{\|{}^E\mathbf{p}_k^j - {}_w^E\mathbf{T}_k {}_i^w\mathbf{p}_k\|} \right)^T \\ &\quad \times \left({}^E\mathbf{v}_k^j - {}_w^E\mathbf{T}_k {}_i^w\mathbf{v}_k \right) - c\dot{\delta t}_k^j - \mathbf{z}_k^d. \quad (8)\end{aligned}$$

6) Receiver Clock Factor: As indicated in (4) and (6), the receiver's clock drifts are time-varying and affect the measured values of pseudorange and carrier phases. The relationship between the receiver clock drifts at two consecutive time steps of the m th GNSS system (with a time interval of Δt) is given by

$$\delta t_k^m = \delta t_{k-1}^m + \dot{\delta t}_{k-1} \Delta t$$

and the corresponding factor is formulated as

$$\mathbf{r}(\mathcal{X}_k, \mathbf{z}_k^{cd}) = \delta t_k^m - \delta t_{k-1}^m - \dot{\delta t}_{k-1} \Delta t.$$

7) Clock Rate Factor: The receiver's clock rate affects the measured Doppler shift, as shown in (8). We have the clock rate as a constant, satisfying

$$\dot{\delta t}_k = \dot{\delta t}_{k-1}$$

and its residual is

$$\mathbf{r}(\mathcal{X}_k, \mathbf{z}_k^r) = \dot{\delta t}_k - \dot{\delta t}_{k-1}.$$

Finally, globally referenced poses can be obtained by optimizing the back-end.

C. Fusion of Front-End With Optimized Results From the Back-End

Upon optimizing the factor graph, the pertinent information distilled from the outlier-ridden and noisy GNSS measurements is consolidated within the optimized results of the state \mathbf{x}^f and the extrinsic ${}_w^E\mathbf{T}$. To enhance the front-end's ability to register LiDAR points and maintain a consistent map—particularly in

scenarios involving LiDAR degradation and long trajectories—we propose integrating the back-end's optimized results into the front-end.

1) *State Refinement*: In order to consistently adjust both the current state values and the covariance values in the front-end based on the effective information from GNSS, we propose using the optimized results of the back-end as the system outputs within the EKF framework of the front-end. Specifically, the latest optimized \mathbf{x}^l , a component of the final state \mathbf{x}^f optimized in the current factor graph, is used as the system output of the EKF. This allows for updating both the current state and covariance in the EKF at the sampling time corresponding to the current GNSS observations. The integration of back-end results into the front-end enhances real-time drift-free trajectory estimation and improves performance in scenarios with LiDAR degradation.

2) *Map Update*: The current state values are updated by the optimized values of the back-end. Consequently, this alteration in the current pose should also have an impact on the map structure.

The adjustment in the current pose influences the poses adjacent to it on the timeline, especially those between the last and the current GNSS observations. To modify those neighboring poses, we employ two clamped B-spline curves to parameterize the trajectories of the IMU frame between the last and current GNSS measurements—one for position $\mathbf{p} \in \mathbb{R}^3$ and another for rotation $\mathbf{R} \in \text{SO}(3)$

$$\begin{aligned} {}_i^w \mathbf{p}(t) &= \sum_{j=0}^{j=n} \mathbf{N}_{j,q}(t) \mathcal{P}_j(t) \\ {}_i^w \mathbf{R}(t) &= \sum_{j=0}^{j=n} \mathbf{N}_{j,q}(t) \mathcal{R}_j(t). \end{aligned}$$

Here, $\mathbf{N}_{j,q}$ denotes the base function of the q th order B-spline curve, obtained through the De Boor-Cox recursive algorithm [54]. \mathcal{P}_j and \mathcal{R}_j represent the control points, with n indicating the number of control points. The parameter t signifies time and serves as the knots of the B-spline curves. We opt for evenly-distributed knots for ${}_i^w \mathbf{p}(t)$ and ${}_i^w \mathbf{R}(t)$. The position and rotation estimations at the knot times serve as control points. Subsequently, we fit the B-spline curves using the approach proposed in [55].

Following the optimization of the back-end, the values of the control points undergo modification, thereby revising the parameters of the B-spline curves representing trajectories. Subsequently, the map points, ${}^w \mathbf{p}_m$ are updated by projecting the LiDAR points through the values of the B-spline curves at the corresponding times

$${}^w \mathbf{p}_m^u(t) = {}_i^w \mathbf{p}^u(t) + {}_i^w \mathbf{R}^u(t)({}_l^i \mathbf{T} \cdot {}^l \mathbf{p}_m(t)).$$

Here, ${}_i^w \mathbf{p}^u(t)$ and ${}_i^w \mathbf{R}^u(t)$ represent the updated values of the B-spline curves. ${}_l^i \mathbf{T}$ denotes the precalibrated extrinsic transformation between the IMU and LiDAR, while ${}^l \mathbf{p}_m(t)$ signifies the measured LiDAR point. Consequently, the map structure is adjusted accordingly.

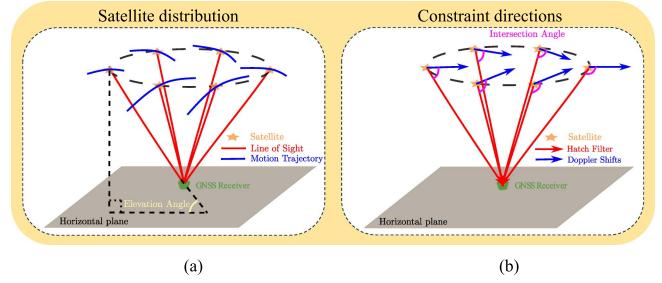


Fig. 3. Illustration for function of Doppler shifts.

The integration of optimized results from the back-end into the front-end's EKF framework is imperative to facilitate real-time globally consistent mapping and global drift-free trajectories.

D. Function Analysis of Three Types of GNSS Observations

First, both Doppler shifts and pseudorange measurements are absolute measurements, incapable of distributing errors among states across different time points to smooth the trajectory. Conversely, the single-difference carrier phase factor establishes a relationship between two consecutive states, offering high accuracy, thereby facilitating local trajectory smoothing.

Subsequently, the Hatch filter furnishes absolute distance measurements with attenuated noises, crucial for estimating globally referenced poses. Refined by more accurate carrier phase measurements, it enhances optimization performance, particularly in challenging scenarios with highly noisy signals.

Finally, Doppler measurements contribute velocity constraints. Furthermore, it augments constraints along directions perpendicular to the Hatch filter measurements for the position estimation of the GNSS receiver, ${}_i^w \mathbf{p}_k$. This augmentation stems inherently from the nature of GNSS systems. Due to the multipath effects, the satellites identified by the GNSS receiver with dependable signals always distribute above the horizontal plane with a significant elevation angle,² denoted as γ . Moreover, the motion trajectories of these satellites are nearly parallel to the horizontal plane, which indicates the velocities of the satellites are also parallel to the horizontal plane. For a visual representation [see Fig. 3(a)]. To address the constraint directions of the Hatch filter (\mathcal{C}_h) and Doppler shifts (\mathcal{C}_d) for position estimation, we calculate the Jacobian matrices of (7) and (8) with respect to the position ${}_i^w \mathbf{p}_k$ [56], as follows:

$$\begin{aligned} \mathcal{C}_h &= \frac{\partial \mathbf{r}(\mathcal{X}_k, \mathbf{z}_k^h)}{\partial {}_i^w \mathbf{p}_k} = \frac{\left({}_w^E \mathbf{T}_k {}_i^w \mathbf{p}_k - {}_E^E \mathbf{p}_k^j \right)^T}{{\| {}_E^E \mathbf{p}_k^j - {}_w^E \mathbf{T}_k {}_i^w \mathbf{p}_k \|}} {}_w^E \mathbf{T}_k \\ \mathcal{C}_d &= \frac{\partial \mathbf{r}(\mathcal{X}_k, \mathbf{z}_k^d)}{\partial {}_i^w \mathbf{p}_k} \end{aligned}$$

²The elevation angle is defined as the angle measured perpendicularly from the Earth's horizontal plane (in the vertical plane) toward the satellite's line-of-sight.

$$= \alpha \frac{\left({}^E \mathbf{v}_k^j - {}_w \mathbf{R}_{k_i} {}^w \mathbf{v}_k \right)^T}{{}^E \mathbf{v}_k^j - {}_w \mathbf{R}_{k_i} {}^w \mathbf{v}_k} {}^E \mathbf{T}_k - \beta \frac{\left({}^E \mathbf{p}_k^j - {}_w \mathbf{T}_{k_i} {}^w \mathbf{p}_k \right)^T}{{}^E \mathbf{p}_k^j - {}_w \mathbf{T}_{k_i} {}^w \mathbf{p}_k} {}^E \mathbf{T}_k \quad (9)$$

where $\beta = \alpha \cos(\theta)$, θ is the intersection angle between two normalized vectors: $\frac{{}^E \mathbf{v}_k^j - {}_w \mathbf{R}_{k_i} {}^w \mathbf{v}_k}{{}^E \mathbf{v}_k^j - {}_w \mathbf{R}_{k_i} {}^w \mathbf{v}_k}$ and $\frac{{}^E \mathbf{p}_k^j - {}_w \mathbf{T}_{k_i} {}^w \mathbf{p}_k}{{}^E \mathbf{p}_k^j - {}_w \mathbf{T}_{k_i} {}^w \mathbf{p}_k}$, and $|\cos(\theta)| \leq |\cos(\gamma)|$. Please refer to the supplementary file [49] for detailed derivations.

As indicated by (9), the constraint direction of the Hatch filter extends from the satellite's position to the receiver's position in the local-world frame. The constraint direction of the Doppler shift is a linear combination of two normalized vectors. The first vector, weighted by α , represents the difference between the satellite and receiver velocities in the local-world frame and is nearly parallel to the satellite's velocity, as the receiver's velocity is negligible in comparison. The second vector, weighted by β , is aligned with the constraint direction of the Hatch filter. Assuming a large elevation angle γ , we have $\beta = \alpha \cos(\theta) \leq \alpha \cos(\gamma) << \alpha$. Therefore, the constraint direction of the Doppler shift is predominantly aligned with the satellite's velocity and intersects the Hatch filter direction at a significant angle, as heuristically illustrated in Fig. 3(b).

VI. EXPERIMENT RESULTS

In this section, we assess the performance of LIGO across various real-world datasets, encompassing both self-collected and publicly available ones.

Aerial vehicles are a vital application requiring GNSS assistance. However, there is a lack of justification results for tightly coupled LiDAR-Inertial-GNSS systems on flight datasets. To address this gap, we utilize self-collected aerial datasets. In addition, we employ a handheld platform to gather data in signal-switching environments, including indoor and outdoor scenarios, to evaluate the robustness of LIGO under varying signal conditions.

For the self-collected dataset, we compare the performance of LIGO against the precise point positioning (PPP) kinematic solution by real-time kinematic library (RTKLIB) [57], Point-LIO [12] with the same map structure [52] as LIGO, as well as a loosely-coupled LIGO variant, denoted as LIGO-loosely. We also compare it against a loosely-coupled LIGO system, Lio-sam-GPS [32]. Point-LIO is reimplemented in this article by utilizing the same map structure [52] of the LIGO. LIGO-loosely is based on Point-LIO [12], utilizing direct registration of LiDAR points to match with the global map [52] and loosely coupling with GNSS position solutions in an EKF framework. Lio-sam-GPS employs a feature-based approach to track between LiDAR scans and fuses GNSS position factors with LiDAR and inertial factors in a graph. We close its loop closure correction to ensure a fair performance comparison. The PPP kinematic solution serves as the position measurement for loosely coupled systems, i.e., LIGO-loosely and Lio-sam-GPS.

For publicly available datasets, we compare LIGO against PPP solution of RTKLIB, LIGO-loosely, and Lio-sam-GPS across all evaluation datasets. In addition, we compare LIGO

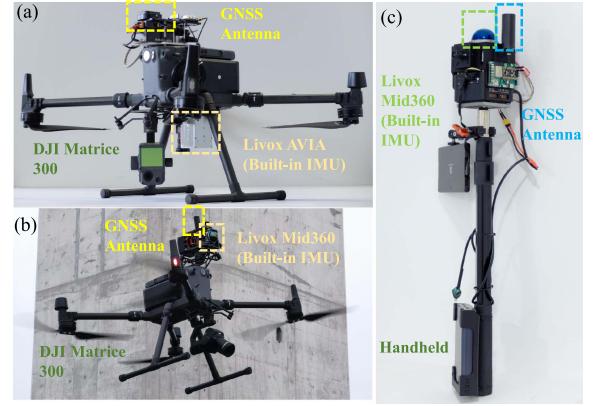


Fig. 4. Platforms for self-collected dataset.

with two tightly coupled GNSS-LiDAR-Inertial systems on suitable sequences: GLIO [22] and another factor-graph-based system [21], referred to as GIL in this article. GLIO fuses double-difference pseudorange and Doppler shift factors with LiDAR and IMU in a graph. Its double-difference pseudorange factors measure the difference of the pseudorange measurements between the GNSS receivers and the base stations. Moreover, GLIO requires a preknown extrinsic of the local-world frame in the ECEF frame to have a globally referenced trajectory estimation. GIL utilizes double-difference carrier phase factors and pseudorange factors to exploit GNSS observations without reliance on base stations. And it is able to estimate global trajectories based on information from pseudorange measurements. However, its code is not publicly available.

In the subsequent experiments, for self-collected sequences using the same LiDAR sensor, we utilize the same set of parameters for LIGO and LIGO-loosely. For all the sequences belonging to the same public dataset, we also deploy the LIGO and LIGO-loosely with the same set of parameters. For quantitative analysis, we compare the global trajectories of all systems with ground truth in the east-north-up (ENU) frame. The estimated trajectories and ground truth are projected to the ENU coordinate using the same anchor point in the ECEF frame. We assess horizon errors considering only the East-North directions, alongside total errors after certain warm-up periods, i.e., 50 s. As the Point-LIO estimation lacks a global reference, we employ ICP using the first 33% of trajectories with the ground truth before comparison. Moreover, for the global reference of GLIO, we use the preknown extrinsic parameters of the local-world frame in the ECEF frame provided in its code to calculate. All trajectory estimations, except Point-LIO, are compared directly with the ground truth without employing ICP.

A. Sensor Set-Ups

For our self-collected dataset, we conducted real-world data collection using three different setups.

A commercial aerial vehicle, DJI Matrice 300, equipped with a Livox AVIA LiDAR and its built-in IMU, as depicted in Fig. 4(a). DJI Matrice 300 equipped with a Livox Mid360 LiDAR and its built-in IMU, as shown in Fig. 4(b). A handheld

platform equipped with a Livox Mid360 LiDAR and its built-in IMU, as illustrated in Fig. 4(c). All three platforms utilize a u-blox ZED-F9P GNSS receiver. Given the payload constraints of the aerial vehicles, the GNSS antennas employed are compact and lightweight. However, they yield lower quality signal observations compared to the commercial-grade antennas typically utilized by grounded platforms.

For all three platforms, the LiDAR measurements are collected at a frequency of 10 Hz, while the IMU data is sampled at 200 Hz. The GNSS raw measurements are acquired at a rate of 10 Hz. To establish ground truth, we utilized a fixed RTK solution obtained by connecting the GNSS receiver to continuously operating reference stations (CORS) via the Internet. The ZED-F9P receiver obtained the Radio Technical Commission for Maritime Services data from the CORS and processed the RTK solution. The RTK solution frequency matches the data collection frequency at 10 Hz.

B. Time Synchronization

In our self-collected data, we achieve synchronization between the LiDAR and GNSS observations using the pulse per second and recommended minimum specific GPS/transit data (GPRMC) message of the GNSS receiver. This synchronization ensures hard synchronization, where the timing of each LiDAR scan aligns precisely with Universal Coordinated Time (UTC). Since the point cloud message and IMU message are already internally synchronized by the LiDAR itself, and the GNSS measurements are aligned with UTC time inherently, all sensors are effectively synchronized.

C. Noise Levels of Measurements

The first challenge encountered when working with aerial vehicles is handling GNSS observations with outlier and noise levels typical of lightweight antennas. We collected a sequence by flying a commercial UAV in a wild field. The UAV followed a predefined trajectory with the “LIGO” formulation in this experiment, and the sequence is denoted as LOGO.

For this sequence, the noise levels³ of the GNSS observations collected for pseudorange, Doppler shifts, and carrier phase are summarized in Fig. 5. The number of signals with different noise levels was divided by the number of GNSS measurements (i.e., 10 Hz) of the LOGO sequence, to calculate the average number of signals per measurement. The pseudorange noise level spans the range of (0.0, 40.0) meters, with the majority of pseudoranges exhibiting noise levels within (0.0, 5.0) meters. In comparison, both Doppler shifts and carrier phase exhibit lower noise levels. The noise level of Doppler shifts is distributed below 2.0 m/s, while that of the carrier phase is below 2.0 m.

The mapping and trajectory estimation results of LIGO are depicted in Fig. 6, demonstrating that LIGO can function effectively under such noise levels. Furthermore, we varied the noise

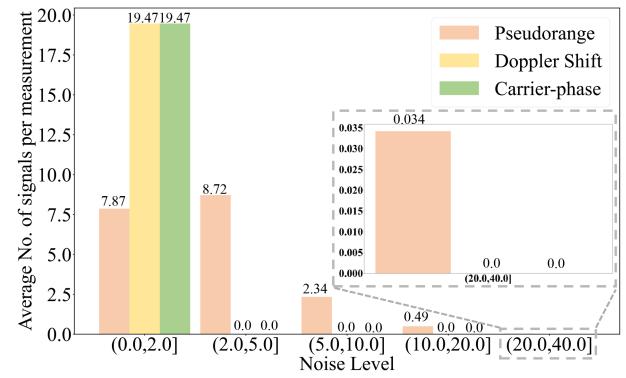


Fig. 5. Noise levels of the GNSS observations. The noise levels for pseudorange, Doppler shifts, and carrier phase have a unit of meters, meters per second, and meters, respectively.

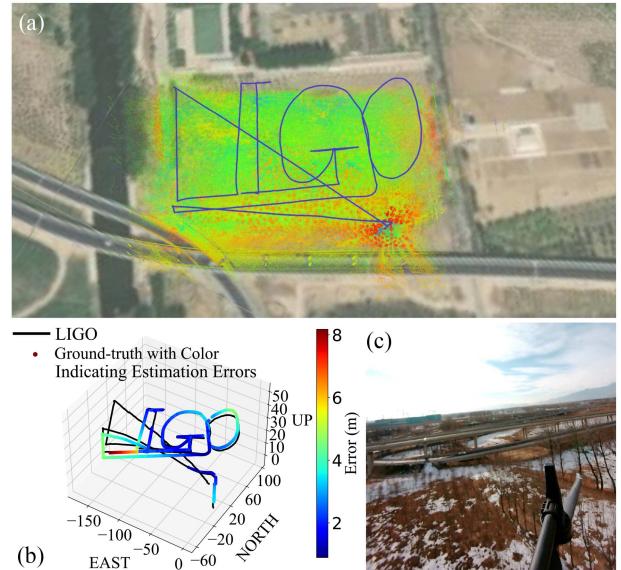


Fig. 6. Trajectory estimation after initial 50 s and mapping results of LIGO for the LOGO sequence.

level thresholds of the pseudorange observations to filter out different regions of observations and compared the root mean squared error (RMSE) of LIGO for different noise level thresholds, as summarized in Fig. 7. Using signals with pseudorange noise level within (0.0, 40.0) meters, LIGO shows the smallest total and horizon RMSEs. However, as shown by the trend of the varying of estimation errors according to different signal levels, the performance of LIGO is not sensitive to this noise level variation of the GNSS signals.

The estimation of the translation and rotation extrinsic parameters of the local-world frame to the ECEF frame is illustrated in Fig. 8. Despite employing a coarse initialization process, both rotation and translation extrinsic parameters converge after a certain time period.

³The value of the noise levels is obtained from the variables “psr_std,” “dopp_std,” and “cp_std” of the GNSS observation messages of the u-blox F9P receivers, for pseudorange, Doppler shift, and carrier phase, respectively. These values represent the standard deviations of the respective observations and are subject to the influence of both observation outliers and noises.

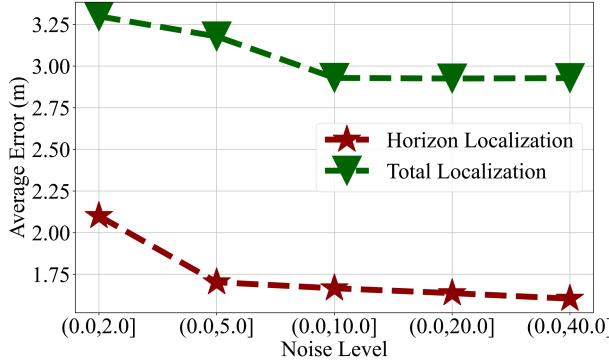


Fig. 7. Localization errors with different noise levels (meters) of the pseudo-range measurements.

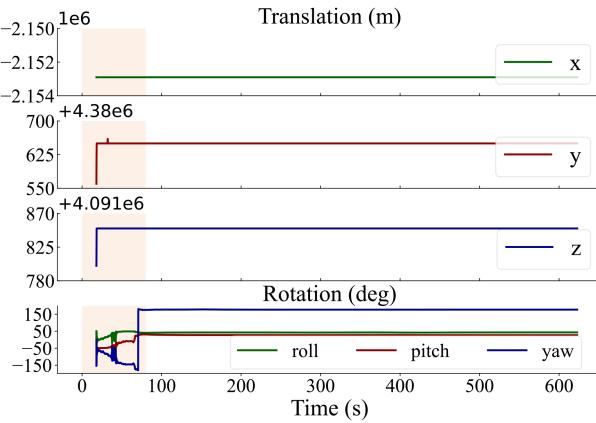


Fig. 8. Estimation of the extrinsic of the local-world frame in the ECEF frame on LOGO sequence. The red background indicates the time period needed for extrinsic to converge.

D. Loss of GNSS Signals

In this section, we evaluate the performance of LIGO on self-collected dataset with loss of GNSS signals involving signal outliers caused by intermittent coverage of GNSS antennae. The dataset includes an indoor–outdoor switch sequence collected by a handheld platform [see Fig. 4(c)], named the switch sequence, and an aerial sequence flying across a bridge underneath carrying a Mid360 LiDAR [see Fig. 4(b)], referred to as the bridge sequence.

1) Switch Sequence: The switch sequence was collected on campus, as depicted in Fig. 9, which shows the surrounding snaps and the trajectory estimation and mapping results in the local-world frame by LIGO. The trajectory includes segments both within the building where the GNSS signals are lost and on the campus streets. The average noise level of the pseudorange signals is 3.33 m. As shown by Fig. 9(c), LIGO produces consistent mapping and trajectory estimations in the local-world frame, and the transferring between indoors [shown by green trajectories in Fig. 9(c)] and the outdoors [shown by red trajectories in Fig. 9(c)] are smooth.

The trajectories after initial 50 s represented in the ENU coordinate of LIGO, LIGO-loosely, Lio-sam-GPS, Point-LIO,

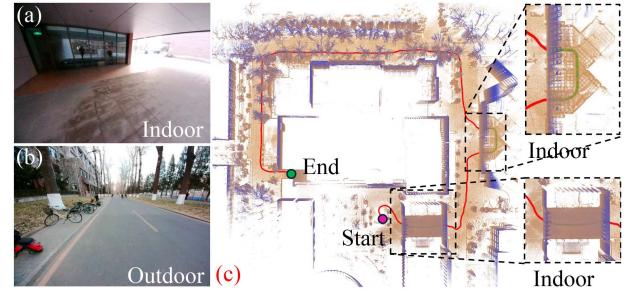


Fig. 9. Mapping results of LIGO on switch sequence. The trajectory with loss of GNSS signals (i.e., indoors) is in green and the remaining outdoor parts are in red. The starting and end points of the trajectory are marked by pink and green dots, respectively.

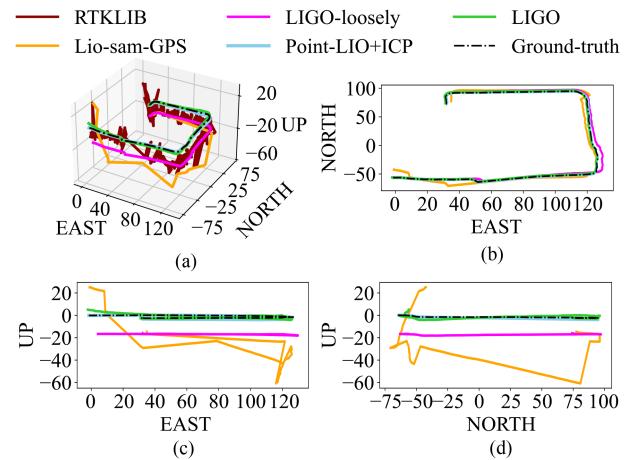


Fig. 10. Comparison of trajectory estimations after the initial 50 s on the Switch sequence. The estimated trajectory of RTKLIB is not presented in (b)–(d).

RTKLIB PPP solutions with ground truth (fixed RTK solution) are presented in Fig. 10. The RMSE results for LIGO, LIGO-loosely, Lio-sam-GPS, Point-LIO, and PPP solutions of RTKLIB are 1.347 m as horizon RMSE with 2.110 m as total RMSE, 3.540 m with 16.04 m, 8.207 m with 42.97 m, 0.4238 m with 0.9795 m, and 5.891 m with 14.25 m, respectively (see Table II as a summarization). As observed, LIGO achieves a horizon RMSE of 1.347 m and a total RMSE of 2.110 m, outperforming other systems in both horizon and total errors, except for Point-LIO, whose RMSEs are 0.4238 m and 0.9795 m, respectively. This superior performance of Point-LIO can be attributed to its alignment with ground truth through a 33% ICP, effectively eliminating errors in the extrinsic estimations. Even though the LIGO shows a discrepancy in meters with the ground truth, this discrepancy is most absorbed in the extrinsic estimation other than the state estimations in the local-world frame, thanks to the hierarchy framework of the LIGO. The PPP solution of RTKLIB exhibits fluctuated errors over time, indicating an unsmooth trajectory estimation. Moreover, due to reliance of the solution of RTKLIB, both LIGO-loosely and Lio-sam-GPS struggle to fuse local LiDAR measurements with the global GNSS position solutions in ENU frame, resulting in large total estimation errors.

TABLE II
COMPARISON OF HORIZON RMSE (M) AND TOTAL RMSE (M) ON SELF-COLLECTED DATASETS

Sequence	Noise	LIGO		LIGO-loosely		RTKLIB		Lio-sam-GPS		Point-LIO+ICP		LIGO-W/O-Dopp		LIGO-W/O-CP	
		Horizon	Total	Horizon	Total	Horizon	Total	Horizon	Total	Horizon	Total	Horizon	Total	Horizon	Total
Switch	3.33	<u>1.347</u>	<u>2.110</u>	3.540	16.04	5.891	14.25	8.207	42.97	0.4238	0.9795	1.393	2.222	1.722	2.762
Bridge	3.67	<u>0.6854</u>	<u>2.581</u>	4.823	10.09	11.20	18.25	8.111	20.60	0.2021	0.5245	0.6995	2.601	0.8926	2.638
Deg-1	2.05	4.020	4.905	-	-	5.042	9.827	-	-	-	-	4.243	5.269	4.105	5.242
Deg-2	2.28	4.920	5.795	-	-	8.040	15.47	-	-	-	-	6.306	8.769	<u>5.360</u>	<u>6.384</u>

¹ The bold and underline text denote the smallest and second smallest RMSE results, respectively. Symbol '-' indicates the values are not available since the system cannot work on the sequence.

² Noise records the average noise level of the pseudo-range measurements in meters.

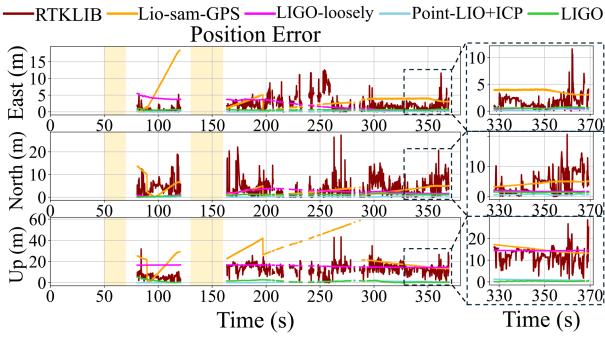


Fig. 11. Trajectory errors over time (after initial 50 s) for the switch sequence. Yellow background indicates the time period with loss of GNSS signals, i.e., indoors.

Fig. 11 illustrates the estimation error over time, focusing only on errors after the initial 50 s and occurring during periods with ground truth (i.e., RTK fixed solution). The yellow background in Fig. 11 shows the time period indoors. The solutions of RTKLIB fluctuate all the time. In addition, both LIGO-loosely and Lio-sam-GPS fail to converge on this sequence. Furthermore, at the end of the sequence, both LIGO and Point-LIO converge, and the steady errors of LIGO are larger than that of Point-LIO along the east and north directions but smaller along the Up direction.

2) *Bridge Sequence*: In the bridge sequence, the DJI Matrice 300 carrying a Livox Mid360 [see Fig. 4(b)] flew underneath a bridge during the flight with an average noise level as 3.67 m. Fig. 12 displays a picture captured under the bridge along with the estimated trajectory and mapping results in the local-world frame obtained using LIGO. As seen from Fig. 12(b), the trajectory in local-world frame is smooth with start and end points coinciding together and the mapping is clear. Moreover, the loss of GNSS signals as denoted by green trajectories does not cause discrepancy on both trajectory estimations and the mapping results.

The 3-D trajectory estimations after initial 50 s along with the ground truth are compared in Fig. 13(a). As shown in Fig. 13, the PPP solutions of RTKLIB are unsatisfactory. To have a clear presentation, we do not show the estimations of RTKLIB in Fig. 13(b)–(d). The horizon RMSE with the total RMSE of

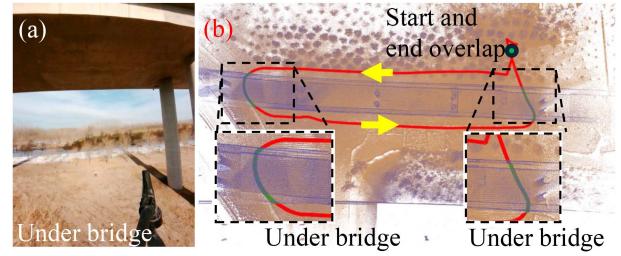


Fig. 12. Mapping result of LIGO on Bridge sequence. The trajectory with loss of GNSS signals (i.e., under bridge) is presented in green and the remaining parts are in red. The starting and end points of the trajectory are marked by pink and green dots, respectively, which coincide together.

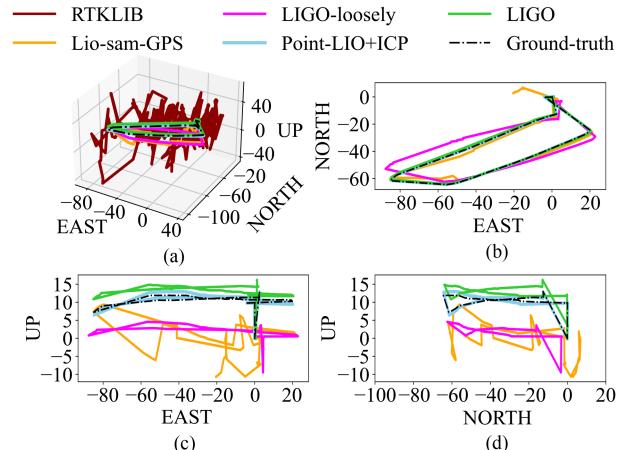


Fig. 13. Comparisons of trajectory estimations after initial 50 s on the Bridge sequence. The estimated trajectory of RTKLIB is not presented in (b)–(d).

LIGO, LIGO-loosely, Lio-sam-GPS, Point-LIO with 33% ICP, and RTKLIB are 0.6854 m with 2.581 m, 4.823 m with 10.09 m, 8.111 m with 20.60 m, 0.2021 m with 0.5245 m, and 11.20 m with 17.86 m, respectively (see Table II for summary). Notably, LIGO achieves the smallest horizon RMSE, i.e., 0.6854 m, and total RMSE, i.e., 2.581 m, among all the systems, except for Point-LIO with 33% ICP (i.e., 0.2021 m and 0.5245 m). Besides the ICP alignment of the Point-LIO, this difference is attributed to the frequent interruption of the GNSS signals, which results

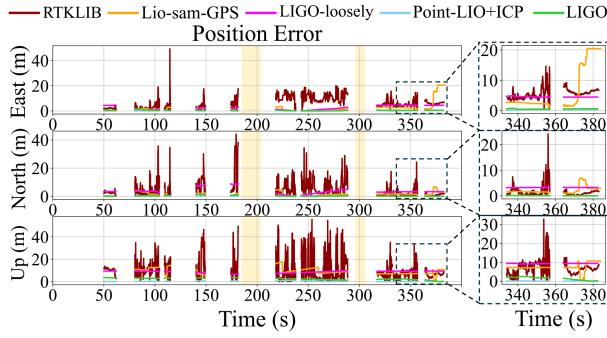


Fig. 14. Trajectory estimation errors over time (after initial 50 s) for the Bridge sequence. Yellow background indicates the time period with loss of GNSS signals, i.e., under the bridge.

in a large discrepancy in the trajectory estimations of the LIGO. Moreover, both LIGO-loosely and Lio-sam-GPS show large discrepancies of trajectory estimations in the ENU frame, due to their reliance on the fluctuated solutions of RTKLIB.

The estimation error over time is illustrated in Fig. 14. It is evident that only half of the trajectory has a fixed RTK solution. In addition, due to the sparse environment structures, the estimation of Lio-sam-GPS shows the largest steady errors at the end of the sequence. The errors at the end of this sequence of LIGO are close to that of Point-LIO along all the east, north, and up directions,

E. LiDAR Degeneracy

In this section, the performance of LIGO on self-collected dataset with LiDAR degeneracy is assessed. The dataset includes two wild-field sequences with severe LiDAR degeneracy collected by the aerial vehicle with an AVIA LiDAR [see Fig. 4(a)], denoted as the Deg-1 sequence and Deg-2 sequence.

1) Deg-1 Sequence: In the first experiment conducted in a wild field with severe LiDAR degeneracy, a Livox AVIA LiDAR was attached to the DJI Matrice 300 facing downwards, as depicted in Fig. 4(a). During this experiment, the aerial vehicle flew above a flight site surrounded by forest and repeatedly experienced low-altitude flights with the LiDAR only scanning a flat ground devoid of any other structures.

Fig. 16 presents the environment pictures captured by a downward camera fixed on the DJI Matrice 300, along with the trajectory and mapping results obtained using LIGO in the first experiment. The aerial vehicle followed a predetermined trajectory, starting at the flight site [indicated by the red dot in Fig. 16(e)] with LiDAR degeneracy. Subsequently, it navigated broadly around the area before returning to the site for low-altitude flights with severe LiDAR degeneracy. This trajectory was repeated three times, with each return to the flight site involving close-to-ground flights where the LiDAR scanned the flat ground only. The LiDAR degeneracy occurs four times during the experiment, and the average pseudorange noise level in this sequence is 2.05 m.

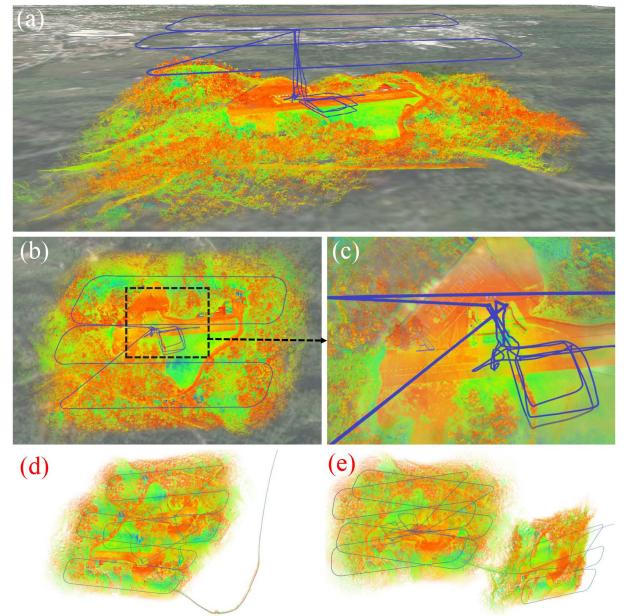


Fig. 15. (a)–(c) Mapping results of LIGO, (d) LIO-loosely, and (e) Point-LIO. (a) and (b) are side and top view, respectively. (c) Local area of (b), most of which is the flat ground. As shown in (d) and (e), severe degeneracy of LiDAR makes LIO-loosely and Point-LIO drift away.

Due to the pronounced degeneracy observed at both the outset and midway through the trajectory, systems such as LIGO-loosely, Lio-sam-GPS, and Point-LIO experienced divergence, as depicted in Fig. 15. Lio-sam-GPS, reliant on features extracted from the current LiDAR scan, initially faltered due to feature loss, leading to errors at the start of the sequence. Conversely, LIGO demonstrated resilience in operating effectively under severe LiDAR degeneracy conditions, consistently maintaining a drift-free trajectory estimation and ensuring coherent mapping in the local-world frame, as showcased in Fig. 16(d).

Fig. 17 illustrates the estimation trajectories after the initial 50 s of both LIGO and the PPP solution of RTKLIB in comparison to the ground truth. Notably, LIGO exhibits a horizon RMSE of 4.020 m and total RMSE of 4.905 m (see Table II for summary), showcasing superior performance over RTKLIB, which demonstrates RMSE values of 5.042 m for horizon and 9.827 m for total, despite encountering repeated severe LiDAR degeneracy. The error shown in the trajectory estimation of LIGO in ENU frame lies in the extrinsic estimation and hardly influences the estimation as well as mapping results in the local-world frame.

Fig. 18 displays the estimation errors over time after the initial 50 s. The red background indicates the occurrence of LiDAR degeneracy. For the time period with severe LiDAR degeneracy, the estimation error of LIGO does not increase, working normally and smoothly under severe LiDAR degeneracy. Moreover, the LIGO shows great convergence at the end of the sequence. This reveals LIGO's ability to achieve converged and smooth estimation results for this sequence.

2) Deg-2 Sequence: This experiment mimics the Deg-1 sequence's collection procedure but with a manually controlled

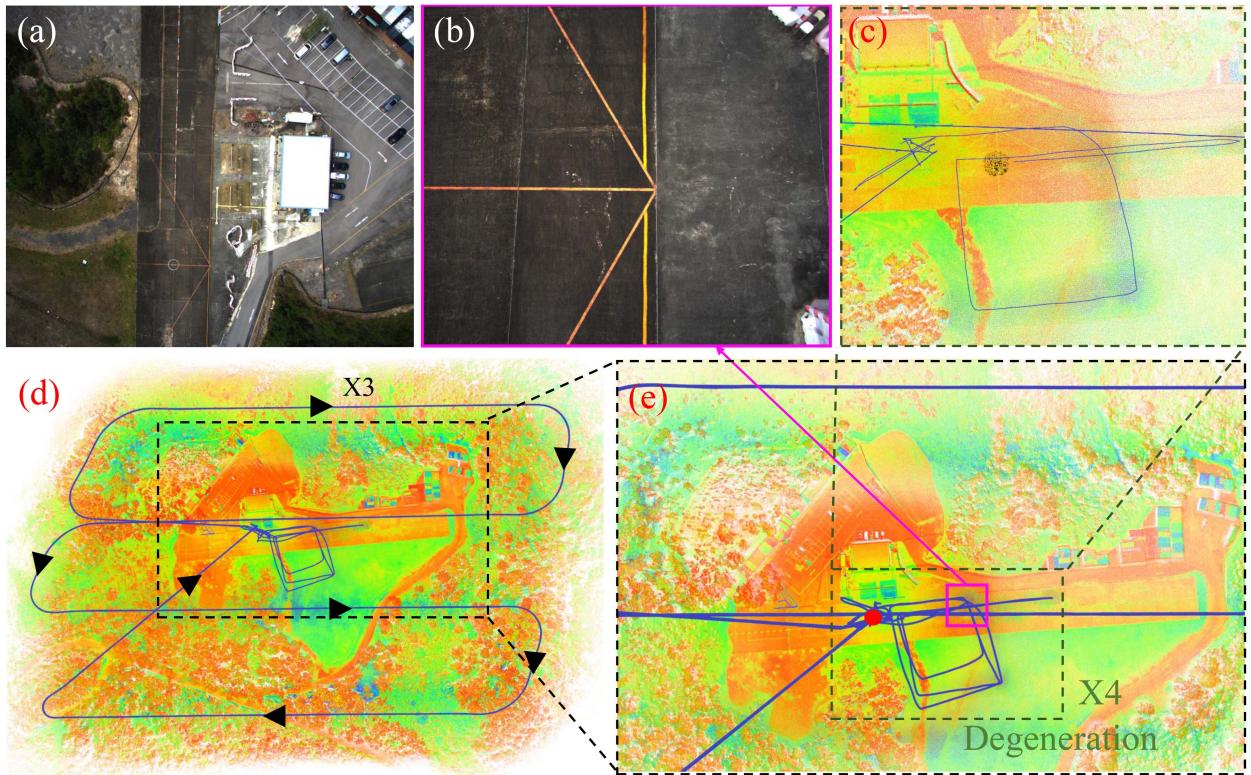


Fig. 16. Mapping results of LIGO on the Deg-1 sequence (c)–(e). A red dot in (e) marks the starting point of the trajectory. The UAV follows the predefined trajectory 3 times. In the area boxed in pink in (e), severe LiDAR degeneracy occurs when scanning flat ground (b). The UAV flies at a low altitude above this flat ground 4 times. (c) Current LiDAR scan in black when scanning the flat floor.

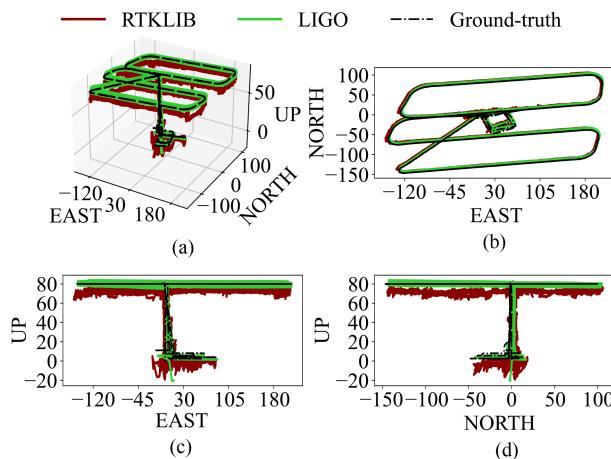


Fig. 17. Comparisons of trajectory estimations after the initial 50 s of LIGO, RTKLIB, and the ground truth. Note that all the LIGO-loosely, Lio-sam-GPS, and Point-LIO diverges on Deg-1 sequence.

arbitrary path instead of a predefined one. Like Deg-1, the UAV initiates from a flat region within the forest where LiDAR degeneracy exists, traverses around, and returns to the flat region five times. Each return triggers severe LiDAR degeneracy. The sequence shows an average 2.28 m pseudorange noise level.

Similar to the Deg-1, only LIGO yields consistent mapping and drift-free trajectories, while LIGO-loosely, Lio-sam-GPS, and Point-LIO fail. Moreover, LIGO's horizon and total RMSEs are 4.920 m and 5.795 m, respectively, smaller than RTKLIB

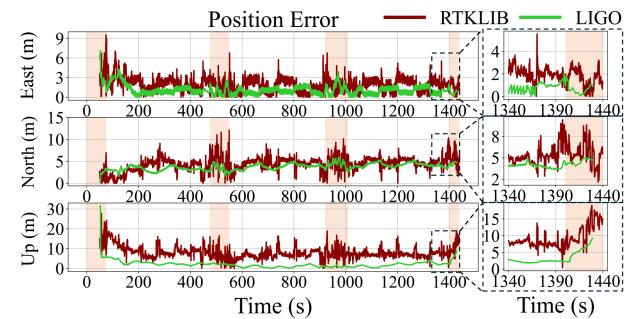


Fig. 18. Estimation errors of trajectory over time (after the initial 50 s) on Deg-1 sequence. Red background indicates time period with LiDAR degeneracy.

PPP solutions with 8.040 m (horizon) and 15.47 m (total). Due to the space limit, refer to the supplementary file [49] for detailed experiment results.

F. Ablation Study

To systematically evaluate the contributions of individual GNSS observational components, namely pseudorange, Doppler shift, and carrier phase, we undertook an ablation study. This involved the development of two modified versions of the LIGO algorithm: one excluding Doppler shift measurements, designated as LIGO-W/O-Dopp, and another excluding carrier phase data, designated as LIGO-W/O-CP. The performance of these variants is meticulously assessed using the self-collected sequences presented in Sections VI-D and VI-E. In addition,

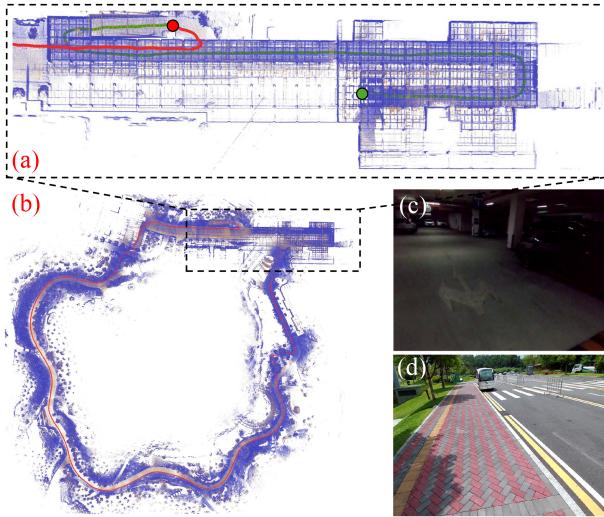


Fig. 19. Mapping results of LIGO on the Indoor-Start sequence (a), (b). A green dot in (a) marks the starting point of the trajectory. The sequence starts in an underground park slot with walking around the slot for the following 290 s as represented by the trajectory in green. A red dot in (a) denotes the beginning of the outdoor collection, with the following trajectory in red as shown in (b). (c) and (d) Camera pictures in the park slot and outdoors, respectively.

a comparative analysis of the trajectory RMSE is performed against the baseline LIGO algorithm. The results, as summarized in Table II, indicate that while both LIGO-W/O-Dopp and LIGO-W/O-CP successfully processed all test sequences, their positional accuracy was compromised in comparison to the original LIGO algorithm. This suggests that the inclusion of both Doppler shift and carrier phase information results in a positive impact on the accuracy and the stability of localization results.

G. Indoor Launch and Drift Mitigation

In this section, we assess the performance of the LIGO under conditions when the initial segment of the test sequence is conducted indoors. The test sequence, referred to as indoor-start sequence, was generated by simultaneously walking and manually operating a sensor unit, as depicted in Fig. 4(c). The experiment involved initiating the data collection from an underground parking garage located within a city garden, an environment characterized by the absence of GNSS signals.

Fig. 19 illustrates the transition from an indoor environment, depicted in Fig. 19(c), to an outdoor setting shown in Fig. 19(d), captured by a forward-facing camera affixed to the handheld platform. Fig. 19 also presents the corresponding trajectory and mapping results generated by the LIGO system as shown by Fig. 19(a) and (b). The experiment commenced within an underground parking garage, an environment devoid of GNSS signals, situated beneath a city garden. The initial 290 s of data collection involved navigating the underground parking area, as represented by the green trajectory in Fig. 19(a). Subsequently, the data acquisition process continued for an additional 960 s, involving ascending to the outdoor environment and extensively exploring the garden without retracing the trajectory, as shown by the red trajectory of Fig. 19(b). The entire sequence spanned

TABLE III
COMPARISON OF HORIZON RMSE (M) AND TOTAL RMSE (M) ON THE INDOOR-START SEQUENCE

	LIGO	LIGO-OUT	LIGO-400	LIGO-900
Horizon	7.384	6.112	5.004	4.979
Total	9.356	7.811	5.014	4.988

a duration of 1250 s, with the average pseudorange noise level characterized as 1.90 m.

1) *Ability of Indoor Launch:* To demonstrate the capability of the LIGO system for indoor launch, a comparative analysis of system's performance was conducted. This assessment involved evaluating LIGO's operation from the inception of the sequence and after a 290-s interval, termed LIGO-OUT, which simulates an outdoor commencement. The first and second columns of Table III exhibit RMSE estimates for both LIGO and LIGO-OUT. The findings reveal that LIGO-OUT surpasses LIGO in both horizon and total RMSEs, suggesting that while LIGO is capable for indoor initiation, its precision is somewhat diminished relative to an outdoor deployment.

2) *Ability of Drift Mitigation:* Using the indoor-start sequence, characterized by a circular trajectory with overlaps occurring only at the initial segments, we evaluate the performance of LIGO in mitigating drift under this challenging scenario. In the experiment, we first apply LIGO to the indoor-start sequence with GNSS observations available only between 290 and 400 s, after which the system fuses LiDAR and IMU measurements (denoted as LIGO-400). As a comparison, LIGO is also run on the same sequence with GNSS available during two periods: 290 to 400 s and 900 to 1250 s (denoted as LIGO-900). In the LIGO-400 configuration, drift occurs after 400 s due to the local spatial constraints from the fusion of LiDAR and IMU measurements in the absence of trajectory overlaps. Similarly, for LIGO-900, drift emerges after 400 s but is mitigated after 900 s when GNSS becomes available once again.

We evaluate the trajectories of LIGO-400 and LIGO-900 within the local-world frame. The initial 20% of the estimated trajectories are aligned with the ground truth using ICP. As indicated in the last two columns of Table III, the reduced horizon and total RMSEs of LIGO-900 compared to LIGO-400 confirm LIGO's ability to mitigate trajectory drift. However, LIGO is unable to fully eliminate the accumulation of drift over time.

H. Benchmark Validation

We test the performance of LIGO on public datasets, including three bags of UrbanNav [58], four sequences of M2DGR-PLUS [59], and an open-sourced sequence of GIL [21]. The UrbanNav is collected on a commercial car, Honda Fit, with high-speed motions and long transparent distances in urban canyons suffering from GNSS outliers caused by multipath effects. It carries a suit of sensors to record the dataset. We chose the Velodyne HDL-32E LiDAR, Xsense MTi-10 IMU, and u-blox ZED-F9P GNSS receiver for validation. Note that

TABLE IV
INFORMATION OF BENCHMARK DATASETS

		UrbanNav	M2DGR	GIL
LiDAR	Type	Velodyne HDL-32E	RoboSense RS-Helios-16	HESAI XT32
	Frequency	10 Hz	5 Hz	10 Hz
IMU	Type	Xsens MTi-10	Built-in IMU of Intel RealSense D435i	Bosch BMI0856
	Frequency	400 Hz	200 Hz	200 Hz
GNSS	Receiver Type	u-blox ZED-F9P, u-blox EVK-M8T		u-blox C099-F9P
	Measure Frequency	1 Hz	1 Hz	5 Hz
	RTK Frequency	1 Hz	100 Hz	5 Hz
Platform	Type	Commercial car	Self-designed quadricycle	Commercial robot dog

TABLE V
DETAILED CHARACTERS OF BENCHMARK SEQUENCES

Sequence	Duration	Features	Noise
UrbanNav-Tst	785 s	Medium urban canyon	- ¹
UrbanNav-Wha	1536 s	Deep urban canyon	-
UrbanNav-Mon	3367 s	Harsh urban canyon	-
M2DGR-Tre	160 s	Campus with tree cover	2.26 m
M2DGR-Swi	292 s	Indoor-outdoor	6.48 m
M2DGR-Bri	501 s	Campus building	3.67 m
M2DGR-Par	149 s	Parking lot	1.16 m
GIL	1122 s	Forest	5.07 m

¹ Noise records average noise level of pseudo-range measurements.

² - denotes the missing of the value.

since there is a malfunction of u-blox ZED-F9P for the sequence UrbanNav-Tst, we use data of EVK-M8T instead. The sequences of M2DGR are recorded using a self-designed quadricycle moving slowly on campus. These sequences encompass diverse GNSS outlier levels, occurring under good sky visibility as well as in areas with coverage by trees or buildings. It also has a set of different sensors, among which we use the RoboSense RS-Helios-16 3D LiDAR, built-in IMU of Intel RealSense D435i, and u-blox ZED-F9P for the experiment. The dataset of GIL was collected on a legged robot dog, Boston dynamic spot, walking in the forest, involving GNSS outliers caused by antenna coverage from dense trees. It uses a HESAI XT32 LiDAR, a Bosch BMI0856 IMU, and a u-blox C099-F9P for collecting data. Details regarding these datasets are summarized in Table IV, with additional comprehensive information for each utilized sequence provided in Table V. Across all datasets, only six channels of the IMU are utilized, comprising 3-axis accelerations and 3-axis angular velocities. For each of those three datasets, i.e., UrbanNav, M2DGR-PLUS, and GIL, the

same set of parameters of LIGO and LIGO-loosely are deployed for all the sequences.

We compare the performance of LIGO with RTKLIB, LIGO-loosely, and Lio-sam-GPS across all datasets. In addition, we contrast LIGO with GLIO using the sequences from GLIO's paper [22], i.e., UrbanNav-Tst and UrbanNav-Wha. It is important to note that GLIO relies on GNSS base station observations and necessitates preknown initial global poses of the receiver in the ECEF frame, rendering it inapplicable to other datasets. Furthermore, we evaluate LIGO against GIL using its publicly available sequence. The evaluation metrics for GIL remain consistent with those presented in its article, as the GIL code is not publicly accessible.

The horizon and total average translation errors (ATE) for all compared systems are summarized in Table VI. Notably, LIGO demonstrates consistent and superior accuracy performance across all sequences, outperforming LIGO-loosely, RTKLIB, Lio-sam-GPS, and GIL on all the sequences. While LIGO presents larger ATEs in UrbanNav-Tst and UrbanNav-Wha as compared to the GLIO. It is worth mentioning that the GLIO only estimates the states in a local-world frame, and does not estimate the extrinsics of local-world frame in the ECEF frame. The global trajectories of GLIO used for ATE calculation are computed using the translation and rotation extrinsics of the local-world frame in the ECEF frame provided in the code of GLIO. Moreover, GLIO relies on pseudorange observations of the base station to construct double-difference pseudorange factors to achieve the state estimation. This suggests GLIO's suitability as a postprocessing system rather than an online one.

Across all sequences, online estimation of the extrinsic parameters of the local-world frame and providing a globally referenced pose present significant sources of estimation errors. The insufficient motion excitations that the benchmark datasets are all collected on a grounded platform with sparse motions along the direction vertical to the ground, impose more challenges on extrinsic estimations. Lio-sam-GPS demonstrates greater absolute trajectory errors (ATEs) compared to PPP solutions from RTKLIB in 6 out of the examined sequences, attributable to the delayed and divergent alignment of extrinsic parameters

TABLE VI
COMPARISON OF HORIZON ATE (M) AND TOTAL ATE (M) ON BENCHMARK DATASETS

	LIGO		LIGO-loosely		RTKLIB		Lio-sam-GPS		GLIO		GIL	
	Horizon	Total	Horizon	Total	Horizon	Total	Horizon	Total	Horizon	Total	Horizon	Total
UrbanNav-Tst	5.935	<u>8.051</u>	8.014	11.12	23.03	34.23	8.061	18.93	4.340*	7.838*	-	-
UrbanNav-Wha	<u>8.506</u>	16.80	15.08	24.11	24.64	39.47	10.46	22.71	2.07*	5.967*	-	-
UrbanNav-Mon	9.484	17.60	<u>17.54</u>	<u>22.49</u>	105.2	174.8	502.1	515.2	-	-	-	-
M2DGR-Tre	1.671	1.951	4.639	<u>8.921</u>	<u>4.620</u>	9.816	8.983	25.05	-	-	-	-
M2DGR-Swi	0.5851	0.7417	<u>3.644</u>	<u>14.62</u>	10.53	14.64	14.38	20.72	-	-	-	-
M2DGR-Bri	0.6715	2.369	5.608	16.74	<u>5.529</u>	<u>15.85</u>	11.44	33.03	-	-	-	-
M2DGR-Par	1.319	1.970	7.430	<u>12.69</u>	6.679	14.15	<u>5.952</u>	15.45	-	-	-	-
GIL	1.748	3.477	11.46	<u>27.66</u>	25.69	39.78	33.23	38.67	-	-	<u>2.07</u>	-

¹ The bold text denotes the smallest ATEs, and the underlined text represents the second smallest ATE results. Symbol ‘-’ indicates the values are not available.

²* denotes the method that uses extra off-board information, including the pseudo-range measurements of the base stations and the pre-calibrated extrinsic of the local-world frame in the ECEF frame.

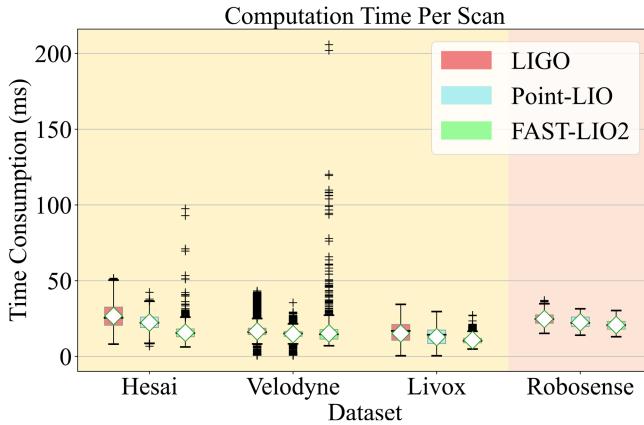


Fig. 20. Computation time comparison of LIGO, Point-LIO with grid map, and FAST-LIO2 [2]. The yellow and red backgrounds represent the sequences with 10 Hz and 5 Hz LiDAR messages, respectively. The rhombuses and the lines within the boxes denote mean and median values, respectively.

between the local-world and the ENU frames. Notably, PPP solutions of RTKLIB also surpasses LIGO-loosely in the M2DGR-Tre and M2DGR-Bri sequences, attributed to the inability to estimate the extrinsics properly of loosely-coupled systems.

I. Computation Time

The analysis of computation time evaluates the real-time applicability and computational time consumption of the proposed system, conducted on a desktop system equipped with an AMD Ryzen9 7945hx processor. The computation time of LIGO for processing one scan of LiDAR points is depicted in Fig. 20 and compared to that of Point-LIO with map of [52] and FAST-LIO2 [2]. The box plots show the statistics of the computation time. The rhombuses within the boxes denote the mean values of time consumption and the lines in the middle of the boxes

show the median values of time consumption. Symbols “+” outside the boxes indicate the outliers. Performance across four sequences with varying LiDAR sensors and GNSS observation rates is demonstrated: the GIL sequence with a 10 Hz HESAI XT32 LiDAR and 10 Hz GNSS observations (denoted as Hesai), UrbanNav-Tst sequence with a 10 Hz Velodyne HDL-32E LiDAR and 1 Hz GNSS observations (denoted as Velodyne), the self-collected bridge sequence with a 10 Hz Livox Mid360 LiDAR and 10 Hz GNSS observations (denoted as Livox), and M2DGR-Par sequence with a 5 Hz RoboSense LiDAR and 1 Hz GNSS observations (denoted as Robosense). The yellow background in Fig. 20 denotes the sequences with 10 Hz LiDAR messages, the time duration of one scan is 100 ms. The red background indicates the sequence with 5 Hz LiDAR messages, whose time period for one scan is 200 ms.

Fig. 20 illustrates that all LIGO and Point-LIO process one LiDAR scan with a time consumption of no more than 100 ms for 10 Hz sequences and no more than 200 ms for 5 Hz sequences, even the largest time overhead, indicating real-time performance. FAST-LIO2 also shows the real-time performance, excepting the 9 outliers of the sequence collected by the Velodyne. Compared with Point-LIO, the average time consumption of LIGO increases by 1.89–4.31 ms, constituting approximately 9.83%–19.2% of the time consumption of Point-LIO. The increased overhead of LIGO compared to Point-LIO arises from the additional processing of GNSS observations. In addition, compared to FAST-LIO, the average time consumption of LIGO increases by 1.35–11.0 ms, representing approximately 8.86%–70.2% of FAST-LIO’s time consumption. As reported in [2], FAST-LIO consumes less than 30 ms for 10 Hz sequences on a mobile computer with an Intel i7-8550U processor. Thus, it is reasonable to infer that LIGO would achieve real-time performance of no more than 50 ms for 10 Hz sequences on the mobile computer, reaching real-time performance.

VII. CONCLUSION

This article presents LIGO, a multisensor fusion framework designed to integrate LiDAR, IMU, and GNSS data effectively. The key innovation lies in fusing the EKF outputs as compact factors and establishing a hierarchical structure that exploits the complementary strengths of different sensor modalities. Specifically, LIGO employs an EKF as a front-end to process LiDAR and IMU measurements, enhancing bandwidth to support high-dynamic, high-speed motions while prioritizing real-time, globally consistent mapping. Conversely, the factor graph back-end addresses the challenges posed by noisy, outlier-prone GNSS measurements, and the substantial scale disparity between local motions and GNSS data. This back-end mechanism is particularly effective in ensuring globally drift-free trajectories and maintaining mapping consistency, even in cases of LiDAR degradation and large-scale environments. The LIGO's performance has been validated in both aerial and ground vehicle applications, demonstrating robustness in scenarios involving signal loss and LiDAR degradation.

This article proposes a novel multisensor fusion framework but also acknowledges several limitations of the current LIGO system, which can be addressed in future research. First, incorporating a well-designed initialization module, particularly for gravity initialization, would be beneficial. Accurate gravity initialization helps constrain the rotation extrinsic of the local-world frame within the ECEF frame. Second, while the LIGO front-end registers LiDAR points directly into a global map in the local-world frame, enhancing robustness in structure-scarce environments and supporting globally consistent mapping, this approach limits the system's ability to fully exploit GNSS observations for mitigating trajectory drifts. Although the back-end's optimized results are integrated into the front-end via the EKF to progressively update system states and map structures, this process can only correct drifts incrementally and is less effective for handling abrupt, large drifts. Third, while this article provides a time consumption analysis comparing LIGO to Point-LIO and FAST-LIO, further evaluation of LIGO's efficiency on onboard computing systems is needed for the future applications. Lastly, as a global odometry system with real-time mapping capabilities, LIGO offers promising avenues for future research. It could provide novel trajectory options for tasks such as path planning, enabling strategies like flying over ground-level obstacles. Furthermore, the real-time odometry and mapping output from LIGO's front-end enhances the system's ability to detect and avoid obstacles in real time. All components of our system have been made publicly available to support the research community.

REFERENCES

- [1] X. Chen, A. Milioto, E. Palazzolo, P. Giguere, J. Behley, and C. Stachniss, "SuMa++: Efficient LiDAR-based semantic SLAM," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2019, pp. 4530–4537.
- [2] W. Xu, Y. Cai, D. He, J. Lin, and F. Zhang, "FAST-LIO2: Fast direct LiDAR-inertial odometry," *IEEE Trans. Robot.*, vol. 38, no. 4, pp. 2053–2073, Aug. 2022.
- [3] I. Vizzo, T. Guadagnino, B. Mersch, L. Wiesmann, J. Behley, and C. Stachniss, "KISS-ICP: In defense of point-to-point ICP—Simple, accurate, and robust registration if done the right way," *IEEE Robot. Autom. Lett.*, vol. 8, no. 2, pp. 1029–1036, Feb. 2023.
- [4] J. Behley and C. Stachniss, "Efficient surfel-based SLAM using 3D laser range data in urban environments," in *Proc. Robot.: Sci. Syst.*, 2018, Paper 59.
- [5] H. Wang, C. Wang, and L. Xie, "Intensity scan context: Coding intensity and geometry relations for loop closure detection," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2020, pp. 2095–2101.
- [6] Q. Li et al., "LO-Net: Deep real-time LiDAR odometry," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 8465–8474.
- [7] Y. Pan, P. Xiao, Y. He, Z. Shao, and Z. Li, "MULLS: Versatile LiDAR SLAM via multi-metric linear least square," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2021, pp. 11633–11640.
- [8] A. Segal, D. Haehnel, and S. Thrun, "Generalized-ICP," in *Proc. Robot.: Sci. Syst.*, Seattle, WA, USA, 2009, vol. 2, no. 4, Paper 435.
- [9] M. Magnusson, "The three-dimensional normal-distributions transform: An efficient representation for registration, surface analysis, and loop detection," Ph.D. dissertation, School Sci. Technol., Örebro Universitet, Örebro, Sweden, 2009.
- [10] M. Magnusson, A. Nuchter, C. Lorken, A. J. Lilienthal, and J. Hertzberg, "Evaluation of 3D registration reliability and speed—A comparison of ICP and NDT," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2009, pp. 3907–3912.
- [11] A. Tagliabue et al., "LION: LiDAR-inertial observability-aware navigator for vision-denied environments," in *Proc. 17th Int. Symp. Exp. Robot.*, 2021, pp. 380–390.
- [12] D. He, W. Xu, N. Chen, F. Kong, C. Yuan, and F. Zhang, "Point-LIO: Robust high-bandwidth light detection and ranging inertial odometry," *Adv. Intell. Syst.*, vol. 5, 2023, Art. no. 2200459.
- [13] G. Kim, S. Choi, and A. Kim, "Scan context++: Structural place recognition robust to rotation and lateral variations in urban environments," *IEEE Trans. Robot.*, vol. 38, no. 3, pp. 1856–1874, Jun. 2022.
- [14] C. Yuan, J. Lin, Z. Liu, H. Wei, X. Hong, and F. Zhang, "BTC: A binary and triangle combined descriptor for 3D place recognition," *IEEE Trans. Robot.*, vol. 40, no. 1, pp. 1580–1599, Jan. 2024.
- [15] K. Li, M. Li, and U. D. Hanebeck, "Towards high-performance solid-state-LiDAR-inertial odometry and mapping," *IEEE Robot. Autom. Lett.*, vol. 6, no. 3, pp. 5167–5174, Jul. 2021.
- [16] Z. Zou et al., "LTA-OM: Long-term association LiDAR–IMU odometry and mapping," *J. Field Robot.*, vol. 41, pp. 2455–2474, 2024.
- [17] Y. Li et al., "Toward location-enabled IoT (LE-IoT): IoT positioning techniques, error sources, and error mitigation," *IEEE Internet Things J.*, vol. 8, no. 6, pp. 4035–4062, Mar. 2021.
- [18] X. Li, H. Wang, S. Li, S. Feng, X. Wang, and J. Liao, "GIL: A tightly coupled GNSS PPP/INS/LiDAR method for precise vehicle navigation," *Satell. Navigat.*, vol. 2, pp. 1–17, 2021.
- [19] S. Li, S. Wang, Y. Zhou, Z. Shen, and X. Li, "Tightly coupled integration of GNSS, INS, and LiDAR for vehicle navigation in urban environments," *IEEE Internet Things J.*, vol. 9, no. 24, pp. 24721–24735, Dec. 2022.
- [20] X. Li, H. Yu, X. Wang, S. Li, Y. Zhou, and H. Chang, "FGO-GIL: Factor graph optimization-based GNSS RTK/INS/LiDAR tightly coupled integration for precise and continuous navigation," *IEEE Sensors J.*, vol. 23, no. 13, pp. 14534–14548, Jul. 2023.
- [21] J. Beuchert, M. Camurri, and M. Fallon, "Factor graph fusion of raw GNSS sensing with IMU and LiDAR for precise robot localization without a base station," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2023, pp. 8415–8421.
- [22] X. Liu, W. Wen, and L. T. Hsu, "GLIO: Tightly-coupled GNSS/LiDAR/IMU integration for continuous and drift-free state estimation of intelligent vehicles in urban areas," *IEEE Trans. Intell. Veh.*, vol. 9, no. 1, pp. 1412–1422, Jan. 2024.
- [23] H. Lee and C. Rizos, "Position-domain hatch filter for kinematic differential GPS/GNSS," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 44, no. 1, pp. 30–40, Jan. 2008.
- [24] W. Lee, P. Geneva, Y. Yang, and G. Huang, "Tightly-coupled GNSS-aided visual-inertial localization," in *Proc. Int. Conf. Robot. Autom.*, 2022, pp. 9484–9491.
- [25] T. Suzuki, "GNSS odometry: Precise trajectory estimation based on carrier phase cycle slip estimation," *IEEE Robot. Autom. Lett.*, vol. 7, no. 3, pp. 7319–7326, Jul. 2022.
- [26] C.-M. Chung et al., "Orbeez-SLAM: A real-time monocular visual SLAM with ORB features and NeRF-realized mapping," in *Proc. 2023 IEEE Int. Conf. Robot. Autom.*, 2023, pp. 9400–9406.
- [27] S. Lynen, M. W. Achterlik, S. Weiss, M. Chli, and R. Siegwart, "A robust and modular multi-sensor fusion approach applied to MAV navigation," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2013, pp. 3923–3929.
- [28] T. Moore and D. Stouch, "A generalized extended Kalman filter implementation for the robot operating system," in *Proc. 13th Int. Conf. Intell. Auton. Syst.*, 2016, pp. 335–348.

- [29] Z. Shen, J. Wang, C. Pang, Z. Lan, and Z. Fang, "A LiDAR-IMU-GNSS fused mapping method for large-scale and high-speed scenarios," *Measurement*, vol. 225, 2023, Art. no. 113961.
- [30] J. Gao, J. Sha, H. Li, and Y. Wang, "A robust and fast GNSS-inertial-LiDAR odometry with INS-centric multiple modalities by IESKF," *IEEE Trans. Instrum. Meas.*, vol. 73, no. 1, Jan. 2024, Art. no. 8501312.
- [31] J. Gao, J. Sha, Y. Wang, X. Wang, and C. Tan, "A fast and stable GNSS-LiDAR-inertial state estimator from coarse to fine by iterated error-state Kalman filter," *Robot. Auton. Syst.*, vol. 175, 2024, Art. no. 104675.
- [32] T. Shan, B. Englot, D. Meyers, W. Wang, C. Ratti, and D. Rus, "LIO-SAM: Tightly-coupled LiDAR inertial odometry via smoothing and mapping," in *Proc. IEEE/RSJ Int. Conf. Intell. robots Syst.*, 2020, pp. 5135–5142.
- [33] S. Shen, Y. Mulgaonkar, N. Michael, and V. Kumar, "Multi-sensor fusion for robust autonomous flight in indoor and outdoor environments with a Rotorcraft MAV," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2014, pp. 4974–4981.
- [34] W. Hess, D. Kohler, H. Rapp, and D. Andor, "Real-time loop closure in 2D LiDAR SLAM," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2016, pp. 1271–1278.
- [35] R. Mascaro, L. Teixeira, T. Hinzmann, R. Siegwart, and M. Chli, "GOMSF: Graph-optimization based multi-sensor fusion for robust UAV pose estimation," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2018, pp. 1421–1428.
- [36] T. Qin, S. Cao, J. Pan, and S. Shen, "A general optimization-based framework for global pose estimation with multiple sensors," 2019, *arXiv:1901.03642*.
- [37] P. Pfreundschuh, R. Bähnemann, T. Kazik, T. Mantel, R. Siegwart, and O. Andersson, "Resilient terrain navigation with a 5 DOF metal detector drone," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2023, pp. 10652–10658.
- [38] W. Wu, X. Zhong, D. Wu, B. Chen, X. Zhong, and Q. Liu, "LIO-fusion: Reinforced LiDAR inertial odometry by effective fusion with GNSS/relocalization and wheel odometry," *IEEE Robot. Autom. Lett.*, vol. 8, no. 3, pp. 1571–1578, Mar. 2023.
- [39] J. Nubert, S. Khattak, and M. Hutter, "Graph-based multi-sensor fusion for consistent localization of autonomous construction robots," in *Proc. Int. Conf. Robot. Autom.*, 2022, pp. 10048–10054.
- [40] J. Liu, W. Gao, and Z. Hu, "Optimization-based visual-inertial SLAM tightly coupled with raw GNSS measurements," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2021, pp. 11612–11618.
- [41] S. Cao, X. Lu, and S. Shen, "GVINS: Tightly coupled GNSS-visual-inertial fusion for smooth and consistent state estimation," *IEEE Trans. Robot.*, vol. 38, no. 4, pp. 2004–2021, Aug. 2022.
- [42] T. Qin, P. Li, and S. Shen, "VINS-Mono: A robust and versatile monocular visual-inertial state estimator," *IEEE Trans. Robot.*, vol. 34, no. 4, pp. 1004–1020, Aug. 2018.
- [43] C. Liu, C. Jiang, and H. Wang, "Variable observability constrained visual-inertial-GNSS EKF-based navigation," *IEEE Robot. Autom. Lett.*, vol. 7, no. 3, pp. 6677–6684, Jul. 2022.
- [44] L. Changwu, J. Chen, and W. Haowen, "InGVIO: A consistent invariant filter for fast and high-accuracy GNSS-visual-inertial odometry," *IEEE Robot. Autom. Lett.*, vol. 8, no. 3, pp. 1850–1857, Mar. 2023.
- [45] T. Li, L. Pei, Y. Xiang, W. Yu, and T.-K. Truong, "P³-VINS: Tightly-coupled PPP/INS/visual SLAM based on optimization approach," *IEEE Robot. Autom. Lett.*, vol. 7, no. 3, pp. 7021–7027, Jul. 2022.
- [46] T. Suzuki, "Time-relative RTK-GNSS: GNSS loop closure in pose graph optimization," *IEEE Robot. Autom. Lett.*, vol. 5, no. 3, pp. 4735–4742, Jul. 2020.
- [47] E. Kim, T. Walter, and J. Powell, "Adaptive carrier smoothing using code and carrier divergence," in *Proc. Nat. Tech. Meeting Inst. Navigat.*, 2007, pp. 141–152.
- [48] Q. Zhang, Z. Chen, F. Rong, and Y. Cui, "An improved hatch filter and its application in kinematic positioning with single-frequency GPS," *Measurement*, vol. 146, pp. 868–878, 2019.
- [49] D. He, H. Li, J. Yin, and F. Zhang, "Supplementary materials for LIGO: A tightly coupled LiDAR-inertial-GNSS odometry based on a hierarchy fusion framework for global localization with real-time mapping," 2024. [Online]. Available: https://github.com/Joanna-HE/LIGO/blob/main/Supplementary_Material_for_LIGO.pdf
- [50] C. Forster, L. Carbone, F. Dellaert, and D. Scaramuzza, "On-manifold preintegration for real-time visual-inertial odometry," *IEEE Trans. Robot.*, vol. 33, no. 1, pp. 1–21, Feb. 2017.
- [51] D. He, W. Xu, and F. Zhang, "Symbolic representation and toolkit development of iterated error-state extended kalman filters on manifolds," *IEEE Trans. Ind. Electron.*, vol. 70, no. 12, pp. 12533–12544, Dec. 2023.
- [52] C. Bai, T. Xiao, Y. Chen, H. Wang, F. Zhang, and X. Gao, "Faster-LIO: Lightweight tightly coupled LiDAR-inertial odometry using parallel sparse incremental voxels," *IEEE Robot. Autom. Lett.*, vol. 7, no. 2, pp. 4861–4868, Apr. 2022.
- [53] C. Qin, H. Ye, C. E. Pranata, J. Han, S. Zhang, and M. Liu, "LINS: A LiDAR-inertial state estimator for robust and efficient navigation," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2020, pp. 8899–8906.
- [54] W. J. Gordon and R. F. Riesenfeld, "B-spline curves and surfaces," in *Computer Aided Geometric Design*. Amsterdam, The Netherlands: Elsevier, 1974, pp. 95–126.
- [55] C. Sommer, V. Usenko, D. Schubert, N. Demmel, and D. Cremers, "Efficient derivative computation for cumulative B-splines on lie groups," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 11145–11153.
- [56] F. Pomerleau et al., "A review of point cloud registration algorithms for mobile robotics," *Found. Trends Robot.*, vol. 4, no. 1, pp. 1–104, 2015.
- [57] A. Ferreira, B. Matias, J. Almeida, and E. Silva, "Real-time GNSS precise positioning: RTKLIB for ROS," *Int. J. Adv. Robot. Syst.*, vol. 17, no. 3, 2020, Art. no. 1729881420904526.
- [58] L.-T. Hsu et al., "Hong Kong UrbanNav: An open-source multisensory dataset for benchmarking urban navigation algorithms," *Navigat.: J. Inst. Navigat.*, vol. 70, no. 4, 2023, Art. no. navi.602.
- [59] J. Yin, A. Li, W. Xi, W. Yu, and D. Zou, "Ground-fusion: A low-cost ground SLAM system robust to corner cases," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2024, pp. 8603–8609.



Dongjiao He (Member, IEEE) received the bachelor's degree in power engineering and engineering thermophysics from Southeast University, Nanjing, China, in 2016 and the master's degree in power engineering and engineering thermophysics from Shanghai Jiao Tong University, Shanghai, China, in 2019, and the Ph.D. degree in robotics from the University of Hong Kong, Hong Kong, in July 2024, under supervision of Prof. Fu Zhang.

Her research focuses of the doctoral degree include advanced multisensor algorithms for autonomous navigation, SLAM, and other related fields of UAVs. She is committed to advancing research for autonomous systems, including multisource information fusion, autonomous perception, and strategy decision.



Haotian Li (Graduate Student Member, IEEE) received the B.E. degree in measuring and control technology and instrumentations from Harbin Engineering University, Harbin, China, in 2021. He is currently working toward the Ph.D. degree in robotics with the Department of Mechanical Engineering, University of Hong Kong (HKU), Hong Kong, China.

His current research interests are on robotics, with a focus on unmanned aerial vehicle (UAV) design, and sensor fusion.



Jie Yin received the B.Eng. degree in electronic science and technology and M.Eng. degree in information and communication engineering from Shanghai Jiao Tong University, Shanghai, China, in 2021 and 2024, respectively.

His current research interests include SLAM, robotic learning, and embodied AI.