

LIO-Fusion: Reinforced LiDAR Inertial Odometry by Effective Fusion With GNSS/Relocalization and Wheel Odometry

Wenhong Wu¹, Xunyu Zhong¹, Dongjie Wu¹, Bushi Chen, Xungao Zhong¹, Member, IEEE, and Qiang Liu¹

Abstract—Reliable state estimation is a prerequisite for autonomous robot navigation in complex environments. In this work, we present LIO-Fusion, a reinforced LiDAR inertial odometry system that optimally fuses GNSS/relocalization and wheel odometry to provide accurate and robust 6-DoF movement estimation under challenging perceptual conditions. LIO-Fusion formulates multi-source sensors fusion based on factor graph, allowing a multitude of relative and absolute measurements which may be degraded, disturbed or even inaccessible. In the LIO-Fusion system, online initialization consists of point cloud feature extraction and matching, IMU preintegration, encoder integration, GNSS calibration and prior-map relocalization. Then, its global reinforcement module detects the reliability of GNSS/relocalization to obtain healthy GNSS/relocalization factors, whereas the local reinforcement module uses a sub-factor graph to fuse prior estimation results for the reinforced local odometry factor. Finally, the basic LiDAR/IMU factors, healthy GNSS/relocalization factors and reinforced local odometry factor are jointly used to constrain the system state in the main factor graph such that low-drift odometry under LiDAR degradation can be reliably obtained and corrected globally. We extensively evaluated the real-time LIO-Fusion system by real-world experiments, and compared its performance to other state-of-the-art methods on large-scale datasets collected in the urban and hazardous environments. Results have shown that LIO-Fusion yielded high precision localization and mapping accuracy as well as robustness to sensor failures.

Index Terms—SLAM, Localization, sensor fusion, mapping, robotics in hazardous fields.

I. INTRODUCTION

RECENTLY, resilient mobile robots have been widely used in various applications such as disaster search [1], [2], industrial monitoring [3] and planetary exploration [4]. In these

Manuscript received 15 August 2022; accepted 14 January 2023. Date of publication 27 January 2023; date of current version 7 February 2023. This letter was recommended for publication by Associate Editor B. Englot and Editor J. Civera upon evaluation of the reviewers' comments. This work was supported in part by the Equipment Advance Research Foundation of China under Grant 61405180205 and in part by the Aeronautical Science Foundation of China under Grant 201908068003. (*Corresponding author: Xunyu Zhong*)

Wenhong Wu, Xunyu Zhong, Dongjie Wu, and Bushi Chen are with the Department of Automation, School of Aerospace Engineering, Xiamen University, Xiamen, Fujian 361102, China (e-mail: wuhw@stu.xmu.edu.cn; zhongxunyu@xmu.edu.cn; wudongjiechn@outlook.com; sp1kechen@foxmail.com).

Xungao Zhong is with the School of Electrical Engineering and Automation, Xiamen University of Technology, Fujian 361024, China (e-mail: zhongxun-gao@163.com).

Qiang Liu is with the Department of Psychiatry, University of Oxford, Warneford Hospital, Oxford OX37JX, U.K. (e-mail: qiang.liu@psych.ox.ac.uk).

This letter has supplementary downloadable material available at <https://doi.org/10.1109/LRA.2023.3240372>, provided by the authors.

Digital Object Identifier 10.1109/LRA.2023.3240372

scenarios, GNSS denial, visual perception failure due to darkness and obscurants (e.g., fog, dust and smoke), LiDAR degradation due to self-similar areas and unstable motion (e.g., slipping, hitting and rotating) are common factors that pose severe perception challenges to robot state estimation (i.e., estimating the robot movement) [5].

In terms of state estimation, a visual sensor, which can be easily affected by illumination, obscurants and unstable motion, has poor reliability and stability when used solely. At the same time, LiDAR sensors provide high-fidelity and long-range 3D point-cloud measurements, and have been commonly used for robot state estimation. Typically, LIO methods estimate the ego-motion of a robot by registering LiDAR point cloud and Inertial Measurement Unit (IMU) measurements. However, the performance of LIO methods can be degraded or even drastically compromised in self-similar areas [6]. Another problem is the error accumulates in a non-closed loop. Therefore, LIO systems need to combine other sensors to improve their accuracy and reliability. However, if not combined properly, the indiscriminate fusion of unreliable information can also lead to potential uncertainty for robot state estimation. In this study, we propose LIO-Fusion, a reinforced LIO system, which can perform real-time pose estimation in extreme and perceptual challenging environments, through reliability detection and optimal fusion with different sensors, such as GNSS, relocalization and wheel odometry. Experiments carried out on various datasets (e.g., *XMU Campus*, *NCLT* [7] and *Nebula Odometry* [8] datasets) have shown that LIO-Fusion can achieve higher state estimation accuracy and robustness compared to other state-of-the-art multi-source sensor fusion systems.

A. Related Works

Various systems have been proposed to integrate LiDAR and other sensors. Generally, they can be classified into two groups, namely loosely and tightly coupled methods.

1) *Loosely Coupled LiDAR-Based Odometry*: Loosely coupled odometry methods process LiDAR and other sensor measurements separately and fuse their results at the end. LOAM [9] and LeGO-LOAM [10] initially estimate the pose calculated by IMU measurement for LiDAR point cloud registration. Extended Kalman filter (EKF) is a typical approach to loosely coupled LIO systems. Robot Localization for ROS [11] is a generic system based on EKF, which has realized the loosely-coupled fusion of multi-source sensors such as LiDAR, IMU, GNSS and wheel odometry. Cartographer [12] implements the fusion of GNSS, wheel odometer, LiDAR and IMU based on graph optimization. LOCUS [5] and LOCUS2.0 [8] are based on LiDAR

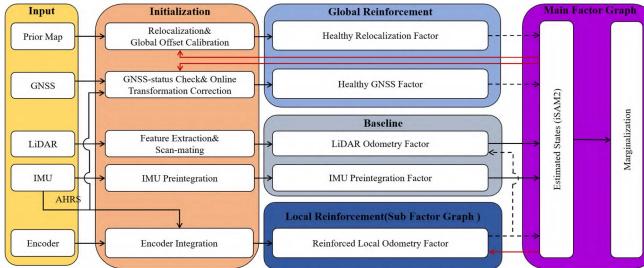


Fig. 1. Architecture of the proposed LIO-Fusion system.

odometry (LO), which used a simple rate-check of sensors as the health-metric-driven dynamic switching to achieve robust positioning under perception challenges. However, the accurate measurements of sensors with low frequency were discarded.

In summary, loosely-coupled methods are flexible to implement and computationally efficient [13]. However, they are easily affected by the processing noise as they fail to correct the internal states of sensors.

2) *Loosely Coupled LiDAR-Based Odometry*: Tightly coupled odometry methods directly fuse LiDAR point clouds with other raw sensor measurements to jointly optimize the state variables. It can be further divided into filter-based and optimization-based methods.

For filter-based methods, iterated extended Kalman filter (IEKF) is often used to reduce estimation errors such as in FAST-LIO [14] and FAST-LIO2 [15]. Compared to FAST-LIO, FAST-LIO2 proposed a data structure incremental k-d tree(ikd-Tree) to search and save maps.

For optimization-based methods, LIO-SAM [16], based on factor graph, constructed a tightly coupled LiDAR-IMU framework with high accuracy and real-time performance. However, it is impossible to be deployed in the presence of sparse geometric features or intermittent failures. LIO-Vehicle [17] extended LIO based on the vehicle dynamics model. However, it is not suitable for heterogeneous robots. In addition, different sensors have different failure scenarios. All fusion methods mentioned above do not have a healthy metric to evaluate potential failures (e.g., GNSS multipath interference, prior maps inadequacy, LiDAR degradation and unstable wheel odometry measurements).

Compared to others, we use the factor graph coupled with healthy checking for appropriately fusing of LiDAR, IMU, Encoder, GNSS and Prior Map to deal with sensor failures. And as far as we know, our work is the first that investigates the effective fusion of multi-source sensors to reinforce the LIO system. Reinforcement means using the constraints of GNSS, Prior Map and Encoder to eliminate the accumulated error of LIO and overcome the LiDAR degeneration, which including global reinforcement outputs healthy GNSS/Relocalization factors and local reinforcement outputs reliable reinforced local odometry factor.

B. Method Highlights and Contributions

1) *Architecture*: The LIO-Fusion architecture (see Fig. 1) is proposed, which enables the effective fusion of sensors according to their current reliability after online initialization. The architecture can be resilient to sensor failure such as LiDAR

degradation and incorrect encoder integration, disturbed/denied GNSS and prior map.

2) *Global and Local Reinforcement for LIO*: The GNSS factor and relocalization factor are fused optimally by healthy check to eliminate the accumulated error. Moreover, through the fusion of prior state estimation results, wheel odometry is enhanced to obtain reliable reinforced local odometry, which can better overcome the LiDAR degradation.

3) *Environmental Robustness*: The comparison of urban and hazardous experiments has shown that LIO-Fusion can estimate states better in cities, indoor and outdoor auditory canals, tunnels and other complex environments, compared with other state-of-the-art methods, such as LOAM [9], FAST-LIO2 [15], LIO-SAM [16], Cartographer [12] and LOCUS2.0 [8].

II. SYSTEM DESCRIPTION

The spatial frames involved in our multi-source sensors fusion system consist of:

Relative sensor frame: Each sensor is attached to a local frame with its reading. In our system, relative sensor frames include the LiDAR frame $(\cdot)^L$, IMU frame $(\cdot)^I$ and encoder frame $(\cdot)^O$. We align the external transformation of IMU and LiDAR, and use the LiDAR frame as our estimation target frame, denoted as the body frame $(\cdot)^B$.

Absolute sensor frame: Absolute sensors include GNSS and the prior map in our system. The single point positioning (SPP) positions using the pseudorange measurements of GNSS are transformed into a semi-global frame, ENU $(\cdot)^G$, in which x, y and z axes point to the east, north and up directions respectively. The scan-matching pose of the global map is generated from the prior map frame $(\cdot)^R$, which is established based on the initial point-clouds of the prior map.

Local world frame: The conventional frame in which LiDAR inertial system operates is represented as the local world frame $(\cdot)^W$. In LIO system, the origin of the local world frame can be arbitrarily fixed and the z axis is often gravity-aligned.

Here, we use $T_z^a = [R_z^a \ p_z^a]^T$ to denote the transformation from frame a to frame z , where $R_z^a \in SO(3)$ and $p_z^a \in \mathbb{R}^3$ represent the rotational part and the translational part, respectively. For the rotational part, the corresponding Hamilton quaternion q_z^a is also used, with \otimes representing multiplication. We use subscript to refer to a moving frame at a specific time. For example, R_{zt}^a represents the rotation from the moving frame a to the fixed frame z at time t .

We define the optimum system state as the situation maximizing a posterior (MAP) given all the measurements of the sensors [13]. States X can be summarized as:

$$\begin{aligned} X &= [x_i \ x_{i+1} \ \dots \ x_j] \\ x_k &= [p_{B_k}^W \ v_{B_k}^W \ q_{B_k}^W \ b_a \ b_g \ T_W^G], k \in [i, j] \end{aligned} \quad (1)$$

where i^{th} is the start index and j^{th} is the end index. The k^{th} state includes position $p_{B_k}^W$, velocity $v_{B_k}^W$, orientation $q_{B_k}^W$, accelerometer bias b_a , gyroscope bias b_g . And T_W^G is the transformation of $(\cdot)^G$ with respect to $(\cdot)^W$.

In this paper, we design the structure of main and sub factor graphs constraining each other. First, in the main factor graph, reinforced global odometry $T_{W_k}^{RG}$ with constraints of Lidar odometry factor, IMU preintegration factor and available healthy GNSS/relocalization factor, reinforced local odometry factor is

used to output smooth and consistent state estimation of the system, and provide prior factor constrains for the sub factor graph. Then, in the sub factor graph, reinforced local odometry $T_{W_\beta}^{RL}$ with the constrains of wheel odometry and reinforced global odometry is got to limit drift of wheel odometry, and provides reliable initial values for LiDAR odometry and process factor constrains for the main factor graph in return.

A. Initialization

As shown in Fig. 1, the initialization module running the point-cloud features extraction & scan match, and IMU preintegration for baseline. And other preprocessing of GNSS, Prior Map and Encoder measurements for reinforcement.

1) *Feature Extraction and Scan Matching:* The edge and planar features are extracted by evaluating the roughness of points over a local region based on [9], and then searching for the local voxel maps M_k at state x_k after filtering to obtain the sub-keyframes. Finally, a new LiDAR scan \mathbb{F}_{k+1} is matched to the local voxel maps M_k via scan-matching to obtain the relative transformation $\Delta T_{k,k+1}$ between x_k and x_{k+1} , which is the LiDAR odometry factor linking these two poses:

$$\Delta T_{k,k+1} = T_k^T T_{k+1} \quad (2)$$

2) *IMU Preintegration:* We follow the IMU preintegration proposed in [18] to estimate the relative movement at interval time Δt :

$$\begin{aligned} \Delta v &= g\Delta t + (a_t - b_t^a - n_t^a)\Delta t \\ \Delta p &= v_t\Delta t + \frac{1}{2}g\Delta t^2 + \frac{1}{2}(a_t - b_t^a - n_t^a)\Delta t^2 \\ \Delta R &= \exp((\omega_t - b_t^\omega - n_t^\omega)\Delta t) \end{aligned} \quad (3)$$

where Δv , Δp and ΔR are the position, velocity and orientation of the IMU preintegration, respectively. v_t is the velocity integrated during the last interval time. a_t and ω_t are the raw IMU measurements at time t . a_t and ω_t are affected by a slowly varying bias b_t^a , b_t^ω and white noise n_t^a , n_t^ω from accelerometers and gyroscopes.

3) *Encoder Integration:* The wheel odometry model is constructed by the encoder integration between LiDAR sub-keyframes. Based on [19], the model combines the linear velocity v_t at time t and the attitude measurement, including roll, pitch and yaw angle. We can integrate the change value of position $\eta_{W_{t,t+\Delta t}}^O \in \mathbb{R}^3$ and orientation $\theta_{W_{t,t+\Delta t}}^O \in \mathbb{R}^4$ at interval time Δt as follows:

$$\begin{aligned} \eta_{W_{t,t+\Delta t}}^O &= R_{W_t}^I v_t \Delta t \\ \theta_{W_{t,t+\Delta t}}^O &= \Delta q_{W_t}^I \end{aligned} \quad (4)$$

where $\Delta q_{W_t}^I$ is the change value of orientation and the rotation matrix $R_{W_t}^I$ is from the corresponding Hamilton quaternion $q_{W_t}^I$ in $(\cdot)^W$ from the Attitude Heading Reference System (AHRS) measurements.

4) *GNSS Status Check and Online Transformation Correction:* The GNSS status measured by less than four satellites are unreliable [13]. Therefore, we drop these GNSS measurements. The GNSS position $p_{G_{\alpha-1}}$ at time $\alpha-1$ is obtained through the longitude, latitude and altitude. The GNSS orientation $R_{G_{\alpha-1}}$ is obtained using AHRS measurements. The GNSS odometry is defined as $T_{G_{\alpha-1}} = [R_{G_{\alpha-1}} p_{G_{\alpha-1}}]^T$. We define $T_{B_{\alpha-1}}$ as the

reinforced global odometry without GNSS factor, and $\hat{T}_{B_{\alpha-1}}^W$ is the reinforced global odometry after adding GNSS factor. And we assume initial global transformation $T_{W_0}^G$ is an identity matrix \mathbf{I} at the beginning. Finally, $T_{W_{\alpha-1}}^G$ at time $\alpha-1$ is corrected by:

$$T_{W_{\alpha-1}}^G = (T_{G_{\alpha-1}})^{-1} T_{B_{\alpha-1}} (\hat{T}_{B_{\alpha-1}}^W)^{-1} \quad (5)$$

5) *Relocalization and Global Offset Calibration:* High-precision prior maps can be obtained by different methods. And relocalizaiton is to regain a constraint from the 3D high-precision prior maps. We first manually choose an initial pose T_{R_0} . Initial LiDAR scan is matched to a sub-Prior map five iterations around the initial pose T_{R_0} using the GICP registration [20]. The zone of the sub-Prior map is $25m^2$. If the transformation epsilon of GICP is less than 0.01 after five iterations, GICP registration has converged and pose \hat{T}_{R_0} can be used as the initial value of relocalization. If the GICP registration fails, we reselect the initial pose T_{R_0} and repeat the above process. The initial reinforced global odometry pose T_{W_0} is obtained from the main factor graph module. Finally, the global offset transformation $T_{W_0}^R$ is calibrated by:

$$T_{W_0}^R = \hat{T}_{R_0} (T_{W_0})^{-1} \quad (6)$$

B. Global Reinforcement

The GNSS/relocalization factor is added to the factor graph as absolute observations to constrain the error of local optimization. Here, we design a novel health metric for global sensor reliability detection.

1) *The Healthy GNSS Factor:* In urban areas, when the robot is surrounded by buildings, GNSS measurements may be incorrect because of the multipath effect [7]. However, the measurements may have good signals. Therefore, the signal noise cannot be used as the only criterion for the health metric. We design a healthy check method based on the distance to solve this problem. Because robotic motion is continuous and the change of position is subtle, the predicted position $\hat{p}_{B_\alpha}^W$ is used to evaluate the GNSS position p_{G_α} at time α . We define the health metric σ_{G_α} as follows:

$$\sigma_{G_\alpha}^2 = (T_{W_{\alpha-1}}^G p_{G_\alpha} - \hat{p}_{B_\alpha}^W)^2, 0 \leq \sigma_{G_\alpha}^2 \leq d_G^2 \quad (7)$$

where the distance confidence threshold d_G is predefined as 5 m. The GNSS residual model $r_G(z_{W_\alpha}^G, X)$ and noise model Ω_{G_α} are constructed as follows:

$$\begin{aligned} r_G(z_{W_\alpha}^G, X) &= \delta \eta_{W_\alpha}^G = T_{W_{\alpha-1}}^G (p_{G_\alpha} - \eta_{G_\alpha}) \\ \Omega_{G_\alpha} &= \sigma_{G_\alpha} (\Omega_{G_\alpha}^{Signal})^{\frac{1}{2}} \end{aligned} \quad (8)$$

where $T_{W_{\alpha-1}}^G$ is corrected from (5), $\eta_{W_\alpha}^G$ is the measurement value of position by GNSS at time α , $\Omega_{G_\alpha}^{Signal}$ is the signal noise of GNSS measurements. Thanks to the constraint of the health metric σ_{G_α} , we can fuse the GNSS factor optimally through the adaptive GNSS covariance.

2) *The Healthy Relocalization Factor:* We obtain the process relocalization odometry $T_{R_\gamma} = [R_{R_\gamma} p_{R_\gamma}]^T$ at time γ based on [21]. Then the relocalization odometry $T_{W_\gamma}^R = [R_{W_\gamma}^R p_{W_\gamma}^R]^T$ is calibrated in $(\cdot)^W$ through $T_{W_0}^R$ from (6) as follows:

$$T_{W_\gamma}^R = T_{W_0}^R T_{R_\gamma} \quad (9)$$

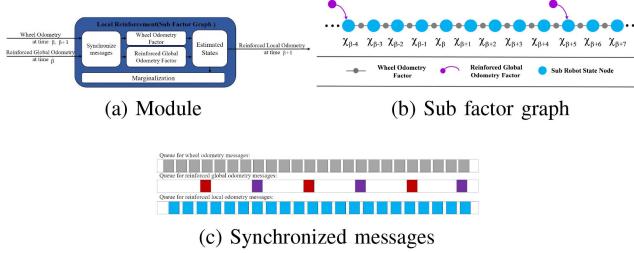


Fig. 2. The structure of local reinforcement (sub factor graph) and process of synchronized messages. In (a), the module does not run in the first time until the first reinforced global odometry is received. In (b) (c), the red messages are synchronized.

The relocalization odometry can be used as the reliable absolute prior information most of the time. However, if a robot explores an unknown environment where the prior map is incomplete, the relocalization odometry may be unreliable in reality. At time γ , we define the distance of the closest points from GICP registration as the health metric σ_{R_γ} as follows:

$$\sigma_{R_\gamma}^2 = d_{R_\gamma}^2, \sigma_{R_\gamma} \leq d_s^2 \quad (10)$$

where d_{R_γ} is the distance of the closest points, and d_s is the distance confidence threshold. In our case, d_s is predefined as 1.0 m. When the prior map is incomplete, d_{R_γ} can construct the weight of the relocalization factor to reduce errors. We then construct the relocalization residual model $r_R(z_{W_\gamma}^R, X)$ and the noise model Ω_{R_γ} as follows:

$$\begin{aligned} r_R(z_{W_\gamma}^R, X) &= [\delta\eta_{W_\gamma}^R \delta\theta_{W_\gamma}^R]^T \\ &= \left[\begin{matrix} p_{W_\gamma}^R - \eta_{W_\gamma}^R \\ [q_{W_\gamma}^R \otimes (\theta_{W_\gamma}^R)^{-1}]_{xyz} \end{matrix} \right] \\ \Omega_{R_\gamma} &= \sigma_{R_\gamma}^2 \mathbf{I} \end{aligned} \quad (11)$$

where $q_{W_\gamma}^R$ is the Hamilton quaternion of $R_{W_\gamma}^R$. $[.]_{xyz}$ is the vector part of the quaternion, \mathbf{I} is the identity matrix, and $[\eta_{W_\gamma}^R \theta_{W_\gamma}^R]^T$ is the relocalization value of position and orientation.

C. Local Reinforcement (Sub Factor Graph)

When robots suddenly accelerate or decelerate, the wheel odometry may drift because of slipping and rotating. In the first-time steps, reinforced global odometry with the constraints of LiDAR odometry and IMU preintegration can be smooth and continue in local scope, which is used as reliable prior constraints added to the sub factor graph (see Fig. 1) to limit the wheel odometry drift. In other words, we combine reinforced global odometry $T_{W_\beta}^{RG}$ with relative wheel odometry $\Delta T_{W_\beta, \beta+1}^O$ to solve wheel odometry slipping. Thus, as shown in Fig. 2(a), we build the module of the local reinforcement (sub factor graph) to receive relative wheel odometry and the closest output of reinforced global odometry ($\tilde{\beta} \leq \beta$ is the closest time to β). Its output is reinforced local odometry. We define the state of the sub-factor graph (see Fig. 2(b)) as follows:

$$\begin{aligned} X_{sub} &= [\chi_a \chi_{a+1} \cdots \chi_b] \\ \chi_\beta &= [p_{W_\beta} q_{W_\beta}], \beta \in [a, b] \end{aligned} \quad (12)$$

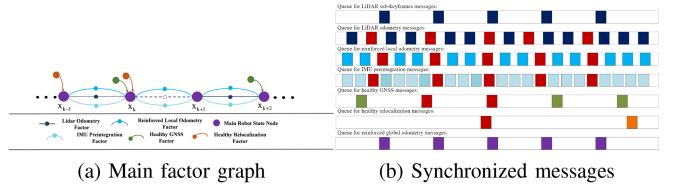


Fig. 3. The main factor graph structure and process of synchronized messages. In (a), the LiDAR odometry factor (dotted line) are not added to the main factor graph when degradation happened. In (b), the red messages are synchronized.

where the β^{th} motion state χ_β includes position p_{W_β} and orientation $q_{W_\beta} a$. We build the sub factor graph from index a^{th} and b^{th} . The wheel residual model is defined as follows:

$$\begin{aligned} r_O(z_{W_\beta, W_{\beta+1}}^O, X_{sub}) &= [\delta\eta_{W_\beta, W_{\beta+1}}^O \delta\theta_{W_\beta, W_{\beta+1}}^O]^T \\ &= \left[\begin{matrix} p_{W_{\beta+1}}^O - p_{W_\beta}^O - \eta_{W_\beta, W_{\beta+1}}^O \\ 2[(q_{W_\beta}^O)^{-1} \otimes q_{W_{\beta+1}}^O \otimes (\theta_{W_\beta, W_{\beta+1}}^O)^{-1}]_{xyz} \end{matrix} \right] \end{aligned} \quad (13)$$

where $[\eta_{W_\beta, W_{\beta+1}}^O \theta_{W_\beta, W_{\beta+1}}^O]^T$ obtained from (3) is the integration value of the wheel odometry, which represents the changes in terms of position and orientation.

To synchronize reinforced global odometry, we interpolate among reinforced global odometry linearly based on the messages of the wheel odometry (See Fig. 2(c)). The synchronized reinforced global odometry residual model is constructed as follows:

$$\begin{aligned} r_{RG}(z_{W_\beta}^{RG}, X_{sub}) &= [\delta\eta_{W_\beta}^{RG} \delta\theta_{W_\beta}^{RG}]^T \\ &= \left[\begin{matrix} p_{W_\beta}^{RG} - \eta_{W_\beta}^{RG} \\ [q_{W_\beta}^{RG} \otimes (\theta_{W_\beta}^{RG})^{-1}]_{xyz} \end{matrix} \right] \end{aligned} \quad (14)$$

where $[\eta_{W_\beta}^{RG} \theta_{W_\beta}^{RG}]^T$ is the predicted value of the reinforced global odometry, which represents the changes in terms of position and orientation. In practice, we only add a reinforced global odometry factor when estimated position covariance is larger than the specified threshold to resist the drift of wheel odometry.

D. Main Factor Graph

Finally, in the main factor graph, we add all reinforced configurations for basic LIO. From the global reinforcement module, the healthy GNSS/relocalization factor is added as the prior constraints to eliminate the accumulated error. From the local reinforcement module, the reinforced local odometry factor enhances the robustness of the system against LiDAR degradation.

As shown in Fig. 3(a), the main factor graph is constructed from the LiDAR, IMU measurements and the reinforced local odometry as tightly coupled constraints. Where, we apply the IMU preintegration from (3) to construct the IMU preintegration residual model between two timesteps [18]. To synchronize reinforced local odometry, we use the timestamp of Lidar sub-keyframes to find the nearest reinforced local odometry and synchronize messages (See Fig. 3(b)). Then, we construct the reinforced local odometry residual model as

follows:

$$\begin{aligned} r_{RL}(z_{W_k, W_{k+1}}^{RL}, X) &= [\delta\eta_{W_k, W_{k+1}}^{RL} \quad \delta\theta_{W_k, W_{k+1}}^{RL}]^T \\ &= \left[\begin{array}{c} p_{W_{k+1}}^{RL} - p_{W_k}^{RL} - \eta_{W_k, W_{k+1}}^{RL} \\ 2 \left[(q_{W_k}^{RL})^{-1} \otimes q_{W_{k+1}}^{RL} \otimes (\theta_{W_k, W_{k+1}}^{RL})^{-1} \right]_{xyz} \end{array} \right] \quad (15) \end{aligned}$$

where $[\eta_{W_k, W_{k+1}}^{RL} \quad \theta_{W_k, W_{k+1}}^{RL}]^T$ is the predicted value of the reinforced local odometry. In addition, the relative transformation $\Delta T_{k,k+1}$ expressed in (2) is used to construct the LiDAR odometry residual model $r_L(m, X)$ [18]. And the reinforced local odometry $T_{W_{k+1}}^I$ and the predicted robot motion $T_{W_{k+1}}^I$ from IMU are used as the initial transformation \hat{T}_{k+1} :

$$\hat{T}_{k+1} = T_{W_{k+1}}^I T_{W_{k+1}}^{RL} \quad (16)$$

When robots explore an environment with sparse geometric features such as long corridors, LiDAR degenerates and LIO fails. Base on the method in [22] for LIO failure detection, we use the best guess of the true state x_p from \hat{T}_{k+1} , $\Delta T_{k,k+1}$ to obtain the final solution x_f . LiDAR relative transformation $\Delta T_{k,k+1}$ is difficult to provide x_p in long corridors because the closest point cloud frames are similar. $T_{W_{k+1}}^I$ diverges because of unreliable forward integration constraints from LiDAR odometry. However, reinforced local odometry $T_{W_{k+1}}^{RL}$ with wheel odometry factor constrains is reliable normally and provide x_p to obtain x_f . The LiDAR odometry factor are not added to the main-factor as well.

E. Factor Graph Optimization

We construct sub and main factor graphs based on the GTSAM framework. The structure is shown in Fig. 2.

1) *Sub-Factor Graph Optimization*: In the sub factor graph, $r_O(z_{W_\beta, W_{\beta+1}}^O, X_{sub})$ and $r_{RG}(z_{W_\beta}^{RG}, X_{sub})$ are added to the following cost function with a Mahalanobis norm as follow:

$$\begin{aligned} X_{sub}^* = \arg \min_{X_{sub}} \left\{ \|r_O(X_{sub})\|^2 \right. \\ + \sum_{\beta \in \{a, \dots, b-1\}} \|r_O(z_{W_\beta, W_{\beta+1}}^O, X_{sub})\|_{\Omega_{O_\beta}^{-1}}^2 \\ \left. + \sum_{\beta \in \{a, \dots, b-1\}} \|r_{RG}(z_{W_\beta}^{RG}, X_{sub})\|_{\Omega_{RG_\beta}^{-1}}^2 \right\} \quad (17) \end{aligned}$$

where $r_O(X_{sub})$ is the prior item form marginalization, Ω_{O_β} is the process noise covariance of the wheel odometry and Ω_{RG_β} is the prior noise covariance of the reinforced global odometry factor.

2) *Main Factor Graph Optimization*: In the main factor graph, $r_L(m, X)$, $r_I(z_{W_k, W_{k+1}}^I, X)$, $r_{RL}(z_{W_k, W_{k+1}}^{RL}, X)$, $r_G(z_{W_\alpha}^G, X)$ and $r_R(z_{W_\gamma}^R, X)$ are added to the following cost function as follow:

$$X^* = \arg \min_X \left\{ \|r_L(X)\|^2 + \sum_{k \in \{i, \dots, j\}} \|r_L(m, X)\|_{\Omega_{I_k}^{-1}}^2 \right\}$$

$$\begin{aligned} &+ \sum_{k \in \{i, \dots, j\}} \|r_I(z_{W_k, W_{k+1}}^I, X)\|_{\Omega_{I_k}^{-1}}^2 \\ &+ \sum_{k \in \{i, \dots, j\}} \|r_{RL}(z_{W_k, W_{k+1}}^{RL}, X)\|_{\Omega_{RL_k}^{-1}}^2 \\ &+ \sum_{\alpha} \|r_G(z_{W_\alpha}^G, X)\|_{\Omega_{G_\alpha}^{-1}}^2 + \sum_{\gamma} \|r_R(z_{W_\gamma}^R, X)\|_{\Omega_{R_\gamma}^{-1}}^2 \} \quad (18) \end{aligned}$$

where $r_L(X)$ is the prior item form marginalization, Ω_{L_k} is the prior noise covariance of the LiDAR odometry, Ω_{I_k} is the process noise covariance of the IMU preintegration, and Ω_{RL_k} is the process noise covariance of the reinforced local odometry. Finally, we optimize the functions using incremental smoothing and mapping with the Bayes tree (iSAM2 [23]) to obtain a stable and reliable state estimation.

III. EXPERIMENTAL RESULTS

In this section, we present the experimental results of LIO-Fusion on *XMU campus*, *NCLT* [7] and *Nebula odometry* [8] datasets. We first used the datasets collected by our inspection robot to perform an ablation study. Compared to other state-of-the-art methods, we show results in the urban and hazardous tests across heterogeneous robotic platforms. See the results¹ and codes² for visualization.

Datasets description: To investigate the impact of each component of LIO-Fusion, we collected the measurements of different sensors at Xiamen University through an inspection robot. Our robot was equipped with a LiDAR (Robosense RS-32/16E, 10 Hz), a low-cost IMU (Xsens MTi-G710, 400 Hz) combined with a GNSS system (UG016, 1 Hz), two-wheel encoders (20 Hz), and a M600mini-G RTK navigation system (to provide the groundtruth, accuracy of 2 cm + 1 um, 1 Hz). Two datasets were collected as *XMU campus*:

- *BuildingX5 Dataset*: The robot moved around a building of Xiamen University for 5 laps with a distance of 1554.68 m.
- *Tunnel Dataset*: The robot passed through a tunnel of Xiamen University with a distance of 276.12 m. There were challenges such as LiDAR degradation and GNSS denied.

The *NCLT* datasets [7] was collected by the Segway robotic platform on the north campus of the University of Michigan. For more details, see [7] and datasets³. We deployed the datasets to simulate robotic inspection in seasonal urban environments. We tested LIO-Fusion in the foliage (2012-04-29 Dataset, 3186.05 m) and snow (2013-01-10 Dataset, 1139.16 m) environments, where messy and slippery roads may cause errors in wheel encoder integration and IMU preintegration.

In order to simulate the inspection in a hazardous environment, we deployed the *Nebula odometry* datasets [8] collected by the CoSTAR team. For more details, see [8] and codes⁴. The ground truth trajectory was produced by running LOCUS2.0 [8] against the ground truth map provided by DARPA. We selected four datasets:

¹[Online]. Available: <https://youtu.be/BB8chVWX7cA>

²[Online]. Available: <https://github.com/wwenhongich/LIO-FUSION>

³[Online]. Available: <http://robots.engin.umich.edu/nclt/>

⁴[Online]. Available: <https://github.com/NeBula-Autonomy/nebula-odometry-dataset>

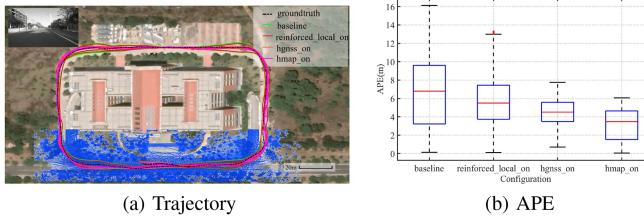


Fig. 4. Localization results and Absolute Position Error (APE) of the proposed method with different processing configurations on *BuildingX5 Dataset* using RTK measurements as the groundtruth. “**baseline**”: the basic tightly-coupled LIO system. “**reinforced_local_on**”: enable local reinforcement (sub factor graph). “**hgnss_on**”: enable the GNSS healthy check. “**hmap_on**”: enable the relocalization heathy check.

TABLE I
LOCAL REINFORCED RMSE(M) W.R.T RTK

Datasets	baseline	reinforced_local_off	reinforced_local_on
2012-04-29	*166.38	22.28	8.85
2013-01-10	6.74	8.46	7.65

¹ Failure leads to a high RMSE, denoted as “*”.

- *Louisville Cave Dataset*: The Husky2 ground rover collected a part of the cave, 325.30 m. The hazardous features of the dataset include darkness, rugged terrain and bumpy motion.
- *Bruceton Mine Pittsburgh Dataset*: The Husky4 ground rover collected a part of the tunnel, 504.55 m. We chose route 2 in the Experimental (EX) course. The perceptual challenges include unstable motion and serious LiDAR degradation.
- *Extra Two Datasets: Satsop Power Plant Elma Dataset* is collected by the Husky4 ground rover, 757.40 m; *Kentucky Underground Limestone Mine Dataset* is collected by the Husky1 ground rover, 1106.29 m.

In order to ensure real-time performance of LIO-Fusion on our robots, we deployed an Ubuntu system with C++, ROS and an Nvidia Jetson AGX Xavier 32 GB.

A. Ablation Study

To investigate the impact of each component of LIO-Fusion on the overall accuracy, we evaluated the final trajectory on the *BuildingX5 Dataset*, where the GNSS measurement was the SPP, and the prior map was only half of the moving area (See the blue map in Fig. 4(a)). The prior maps were firstly built by LIO-Fusion with loop-closure detection. Fig. 4(a) shows the positioning result was smoother and closer to the ground truth when more factor constraints were added. Fig. 4(b) shows the absolute position error (APE), which verified the effectiveness of added factor constraints.

To demonstrate the effectiveness of the local reinforcement stage, we do experiments on the *NCLT* datasets [7] to show how would the LIO-Fusion system performance changes as the local reinforcement enable/disable. Table I shows that “**reinforced_local_on**” was improved in localization compared to “**reinforced_local_off**”. “**baseline**” caused localization and mapping failures on the *2012-04-29 Dataset*. Fig. 5 shows that local reinforcement stage effectively constrained wheel

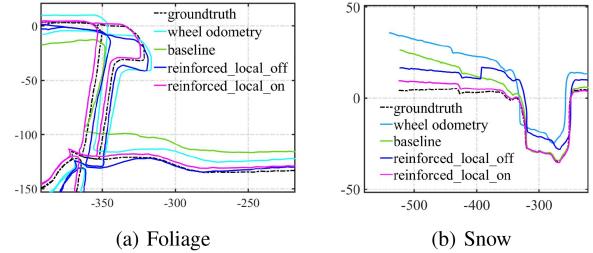


Fig. 5. Localization results of “**baseline**”, “**reinforced_local_off**” and “**reinforced_local_on**” on the corner. “**reinforced_local_off**”: disable the sub factor graph and local reinforcement and directly add the wheel odometry factor to the main factor graph.

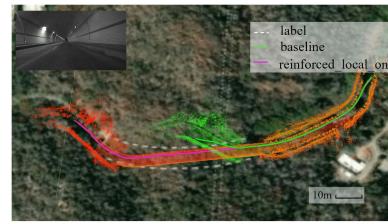


Fig. 6. Localization and mapping result in the *Tunnel Dataset*. “**baseline**” failed because of the LiDAR degradation. However, “**reinforced_local_on**” still obtain the robust localization and mapping result. “label” marked the approximate location of the tunnel in Google map.

TABLE II
GLOBAL REINFORCED RMSE(M) W.R.T RTK

Datasets	hgnss_off	hgnss_on	hmap_off	hmap_on
2012-04-29	10.11	6.53	4.01	3.82
2013-01-10	*32.16	3.62	5.05	3.48

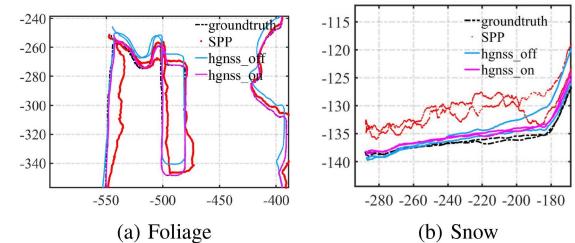


Fig. 7. Localization results of “**hgnss_off**” and “**hgnss_on**” in the environment with unreliable SPP results. “**hgnss_off**”: disable the GNSS healthy check and directly add GNSS factor.

odometry drift on the corner. To demonstrate that “**reinforced_local_on**” can improve robustness, we tested the configuration using the Tunnel dataset. Fig. 6 shows that the system with “**reinforced_local_on**” obtained more robust localization and mapping results even in the LiDAR degraded environment. To demonstrate the effectiveness of the global reinforcement stage, we do experiments on the *NCLT* datasets to show how would the LIO-Fusion system performance changes as the healthy check enable/disable. Table II shows that the configurations with healthy check were improved in localization. Fig. 7 shows that the “**hgnss_on**” eliminated most effect of the unreliable SPP results around buildings. Fig. 8 shows that

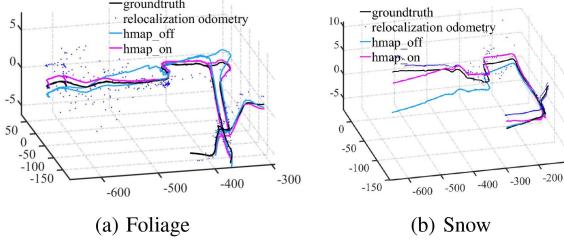


Fig. 8. Localization results of “**hmap_off**” and “**hmap_on**” in the environment with the lack of the large-area Prior maps, which is registered to obtain unreliable relocalization odometry. “**hmap_off**”: disable the relocalization healthy check and directly add relocalization factor.

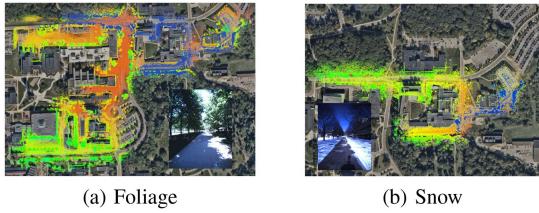


Fig. 9. LIO-Fusion mapping results aligning with satellite map. The blue maps are the Prior maps.

TABLE III
RMSE(M) W.R.T GROUNDDRUTH

Datasets	LOAM	FAST-LIO2	LIO-SAM	Cartographer	LIO-Fusion
2012-04-29	81.93	8.15	*165.32	38.98	3.40
2013-01-10	26.93	5.00	5.13	*118.46	3.10

¹ We do not adjust the parameter settings for comparison methods.

“hmap_on” eliminated most effect of the lack of the large-area Prior maps. The blue point cloud maps in Fig. 9 are prior maps used in the second experiment.

B. Urban Experiment

In the urban environment (See Fig. 9), we demonstrate the accuracy and the robustness of LIO-Fusion in different heterogeneous robots compared with the state-of-the-art methods using the *NCLT* datasets [7]. We tested LOAM [9], FAST-LIO2 [15], LIO-SAM [16] and Cartographer [12]. In Table III, we present the translation root mean square error (RMSE) of the five methods in the foliage and snow environments. LOAM and Cartographer had large accumulated errors. FAST-LIO2 achieved higher localization accuracy in the urban environment. LIO-SAM showed larger drift in Fig. 10(a) due to unreliable initial values and constraints of IMU preintegration added to the back-end factor graph optimization with very long time and long distance [15]. However, LIO-Fusion with reliable reinforced local odometry, healthy GNSS/relocalization can keep smooth and more robust state estimation in terms of all results in Fig. 10. The urban experiments show that LIO-Fusion performed state estimation accurately and robustly in outdoor environments.

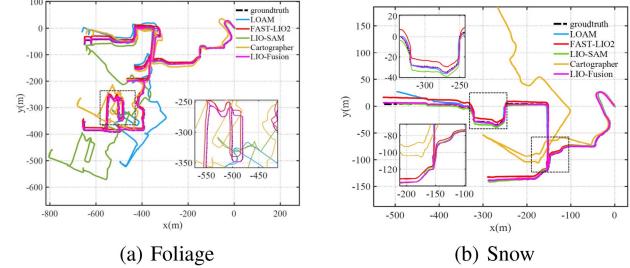


Fig. 10. Localization results of various methods in *NCLT* Datasets.

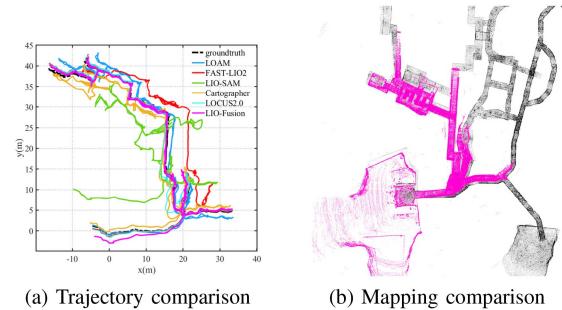


Fig. 11. Localization and mapping result in the *Louisville Cave Dataset*. The black point-clouds map is the groundtruth map, and the pink map is the mapping results of LIO-Fusion.

C. Hazardous Experiment

In these experiments, we evaluated the performance of LIO-Fusion in hazardous environments using the *Nebula odometry* Datasets. In addition, we compared our system with LOCUS2.0 [8]. Specifically, our system used one horizontal mounted LiDAR, IMU and wheel odometry.

In Table IV, we present the results in accuracy. The results show that LIO-Fusion achieved reliable state estimation in hazardous environments and had the smallest mean error (ME) and RMSE. FAST-LIO2 and LIO-SAM had large accumulated errors in cave environment because of the low IMU preintegration frequency (only 50 Hz) in the cave environment. LOAM and Cartographer performed localization accurately in the cave environment. Fig. 11(a) shows the positioning results of six methods and the ground truth using the *Louisville Cave Dataset*. The LIO-Fusion positioning results are smooth and closest to the ground truth. We also compared the mapping result of LIO-Fusion with the ground truth map (see Fig. 11(b)) to show its high-precision performance.

Fig. 12(a) shows the planar positioning results of six methods and the ground truth. Fig. 12(b) presents the altitude positioning results, which are the changes in z-axis. The experimental results show that FAST-LIO2 and Cartographer were most affected by LiDAR degradation. LOAM and LIO-SAM also made large state estimation errors. LOCUS2.0 was still able to perform accurate state estimation on the xy-planer in LiDAR degraded environment because it used multiple LiDARs to add geometric features, but the state estimation of z-axis had a large deviation. Fig. 12(c) shows good localization and mapping performance of LIO-Fusion in the tunnel environment. The extra results could be found in our video due to the limitation of the length.

TABLE IV
SUMMARY OF ACCURACY ANALYSIS RESULTS W.R.T GROUNDTRUTH

Datasets	ME(m)						RMSE(m)					
	LOAM	FAST-LIO2.0	LIO-SAM	Cartographer	LOCUS 2.0	LIO-Fusion	LOAM	FAST-LIO2.0	LIO-SAM	Cartographer	LOCUS 2.0	LIO-Fusion
Louisville Cave	1.59	3.55	9.94	1.52	1.1	0.65	1.76	4.38	10.56	1.94	1.38	0.77
Bruceton Mine Pittsburgh	14.82	-	11.05	-	6.08	1.45	22.55	-	12.42	-	7.9	1.61
Satsop Power Plant Elma	0.57	3.22	5.46	0.89	0.4	0.35	0.74	4.34	7.74	1.05	0.45	0.38
Kentucky Underground Limestone Mine	5.92	6.16	6.55	5.24	12.36	4.87	6.53	6.56	11.28	5.61	16.17	5.52

¹ Methods don not work in the tunnel environment, denoted as “-”.

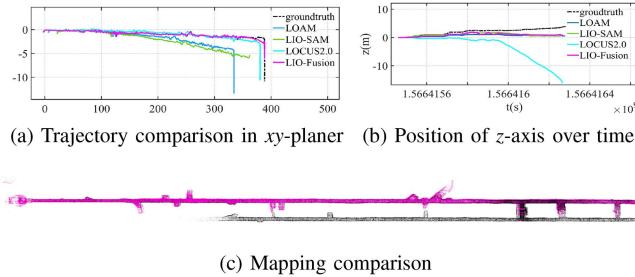


Fig. 12. Localization and mapping result in the tunnel environment.

IV. CONCLUSION

We proposed LIO-Fusion, an optimized fusion system of GNSS/relocalization and wheel odometry, to reinforce LiDAR inertial odometry and mapping in perceptually-challenging conditions. In the initialization stage, the sensors of our system initialized and obtained their transformation. In the global reinforcement stage, reliability detection was introduced to optimize the fusion of the GNSS/relocalization factor to reduce the impact of potential sensor failures. In the local reinforcement stage, we deployed the sub-factor graph and the main factor graph to constrain each other. In the local reinforcement stage, the drifting errors of the wheel odometry were constrained, thus enabling accurate and resilient ego-motion estimation even in the presence of LiDAR degradation. Finally, we compared to other state-of-the-art open-source methods, and LIO-Fusion showed high accuracy and robustness in both urban and hazardous real-world datasets. The proposed architecture is also adaptable to heterogeneous robots in various environments.

REFERENCES

- [1] A. Agha, K. Mitchell, and P. Boston, “Robotic exploration of planetary subsurface voids in search for life,” in *Proc. AGU Fall Meeting Abstr.*, 2019, vol. 2019, pp. P41C-3463.
- [2] T. Sasaki, K. Otsu, R. Thakker, S. Haesaert, and A. A. Agha-mohammadi, “Where to map? iterative rover-copter path planning for mars exploration,” *IEEE Robot. Automat. Lett.*, vol. 5, no. 2, pp. 2123–2130, Apr. 2020.
- [3] V. V. Kostenko and A. Y. Tolstonogov, “The robotic complex for remote aqua monitoring,” in *Proc. OCEANS-MTS/IEEE Kobe Techno-Oceans*, 2018, pp. 1–9.
- [4] J. Zapf, G. Ahuja, J. Papon, D. Lee, J. Nash, and A. Rankin, “A perception pipeline for expeditionary autonomous ground vehicles,” *Proc. SPIE*, vol. 10195, pp. 122–141, 2017.
- [5] M. Palieri et al., “LOCUS: A multi-sensor Lidar-centric solution for high-precision odometry and 3D mapping in real-time,” *IEEE Robot. Automat. Lett.*, vol. 6, no. 2, pp. 421–428, Apr. 2021.
- [6] K. Ebadi, M. Palieri, S. Wood, C. Padgett, and A.-A. Aghamohammadi, “DARE-SLAM: Degeneracy-aware and resilient loop closing in perceptually-degraded environments,” *J. Intell. Robot. Syst.*, vol. 102, no. 1, pp. 1–25, 2021.
- [7] N. Carlevaris-Bianco, A. K. Ushani, and R. M. Eustice, “University of Michigan North Campus long-term vision and Lidar dataset,” *Int. J. Robot. Res.*, vol. 35, no. 9, pp. 1023–1035, 2016.
- [8] A. Reinke et al., “Locus 2.0: Robust and computationally efficient LiDAR odometry for real-time 3D mapping,” *IEEE Robot. Automat. Lett.*, vol. 7, no. 4, pp. 9043–9050, Oct. 2022.
- [9] J. Zhang and S. Singh, “Low-drift and real-time Lidar odometry and mapping,” *Auton. Robot.*, vol. 41, no. 2, pp. 401–416, 2017.
- [10] T. Shan and B. Englot, “LeGO-LOAM: Lightweight and ground-optimized Lidar odometry and mapping on variable terrain,” in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2018, pp. 4758–4765.
- [11] T. Moore and D. Stouch, “A generalized extended Kalman filter implementation for the robot operating system,” in *Proc. Intell. Auton. Syst.*, 2016, vol. 3, pp. 335–348.
- [12] W. Hess, D. Kohler, H. Rapp, and D. Andor, “Real-time loop closure in 2D LiDAR SLAM,” in *Proc. IEEE Int. Conf. Robot. Automat.*, 2016, pp. 1271–1278.
- [13] 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.
- [14] W. Xu and F. Zhang, “FAST-LIO: A fast, robust LiDAR-inertial odometry package by tightly-coupled iterated Kalman filter,” *IEEE Robot. Automat. Lett.*, vol. 6, no. 2, pp. 3317–3324, Apr. 2021.
- [15] 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.
- [16] 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.
- [17] H. Xiao, Y. Han, J. Zhao, J. Cui, L. Xiong, and Z. Yu, “LIO-VEHICLE: A tightly-coupled vehicle dynamics extension of LiDAR inertial odometry,” *IEEE Robot. Automat. Lett.*, vol. 7, no. 1, pp. 446–453, Jan. 2022.
- [18] 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.
- [19] D. Wu, X. Zhong, X. Peng, H. Hu, and Q. Liu, “iSAM2: Incremental smoothing and mapping using the Bayes tree,” *IEEE Trans. Instrum. Meas.*, vol. 71, pp. 1–15, 2022.
- [20] K. Koide, M. Yokozuka, S. Oishi, and A. Banno, “Voxelized GICP for fast and accurate 3D point cloud registration,” in *Proc. IEEE Int. Conf. Robot. Automat.*, 2021, pp. 11054–11059.
- [21] K. Koide, J. Miura, and E. Menegatti, “A portable three-dimensional LiDAR-based system for long-term and wide-area people behavior measurement,” *Int. J. Adv. Robot. Syst.*, vol. 16, no. 2, 2019, Art. no. 1729881419841532.
- [22] J. Zhang, M. Kaess, and S. Singh, “On degeneracy of optimization-based state estimation problems,” in *Proc. IEEE Int. Conf. Robot. Automat.*, 2016, pp. 809–816.
- [23] M. Kaess, H. Johannsson, R. Roberts, V. Ila, J. J. Leonard, and F. Dellaert, “iSAM2: Incremental smoothing and mapping using the Bayes tree,” *Int. J. Robot. Res.*, vol. 31, no. 2, pp. 216–235, 2012.