

# SI-LIO: High-Precision Tightly-Coupled LiDAR-Inertial Odometry via Single-Iteration Invariant Extended Kalman Filter

Cong Zhang, Jie Zhang<sup>✉</sup>, Qingchen Liu<sup>✉</sup>, Yuanzhou Liu<sup>✉</sup>, and Jiahui Qin<sup>✉</sup>

**Abstract**—This letter focuses on the accuracy of LiDAR-inertial odometry (LIO). We propose a novel high-precision tightly-coupled LIO method, SI-LIO, based on the invariant extended Kalman filter with a single-iteration estimate update. This method utilizes the Lie exponential map between the matrix Lie group  $\text{SE}_4(3)$  and its Lie algebra such that the linear approximations referring to Taylor expansions are only conducted on the orientation estimation errors in estimation process. Compared to the state-of-the-art filter-based LIO systems, our method effectively reduces the linearization errors, which enhances the system accuracy, notably in scenarios accompanied by larger prediction errors. Besides, the single-iteration update nature endows SI-LIO with a shorter estimation time than FAST-LIO2. Experiments conducted on public datasets demonstrate the higher accuracy of SI-LIO. Specifically, the estimation accuracy of SI-LIO outperforms FAST-LIO2 by approximately 15% when they both employ a single-iteration update.

**Index Terms**—Kalman filter, Lie groups, odometry, sensor fusion.

## I. INTRODUCTION

DEUE to the high precision, high resolution, long-range detection capabilities, and reliability across various environmental conditions, the light detection and ranging (LiDAR) sensors are popular in simultaneous localization and mapping (SLAM) [1], [2]. However, they also have limitations such as providing insufficient feature information in weakly structured environments and susceptibility to fast motion distortion [3].

Received 17 July 2024; accepted 30 October 2024. Date of publication 13 November 2024; date of current version 11 December 2024. This article 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 National Natural Science Foundation of China under Grant 62425309 and Grant 62403444, in part by the Postdoctoral Fellowship Program of CPSF under Grant GZC20232548, in part by China Postdoctoral Science Foundation under Grant 2023M743386, and in part by the Natural Science Foundation of Anhui Province, China, under Grant 2408085QF192. (Corresponding author: Jiahui Qin.)

Cong Zhang, Jie Zhang, Qingchen Liu, and Yuanzhou Liu are with the Department of Automation, University of Science and Technology of China, Hefei 230027, China (e-mail: Cong\_Zhang@ustc.edu.cn; jiezh@mail.ustc.edu.cn; qingchen\_liu@ustc.edu.cn; lyzzz@mail.ustc.edu.cn).

Jiahui Qin is with the Department of Automation, University of Science and Technology of China, Hefei 230027, China, and also with the Institute of Artificial Intelligence, Hefei Comprehensive National Science Center, Hefei 230088, China (e-mail: jhqin@ustc.edu.cn).

The source code of our LIO system are available at <https://github.com/USTC-AIS-Lab/SI-LIO.git>.

This letter has supplementary downloadable material available at <https://doi.org/10.1109/LRA.2024.3497757>, provided by the authors.

Digital Object Identifier 10.1109/LRA.2024.3497757

[4], which reduce the accuracy of the LiDAR-based SLAM. To deal with this issue, researchers and engineers in relevant fields utilize inertial measurement units (IMUs) to compensate for the shortcomings of LiDAR, designing LiDAR-inertial odometry (LIO) to achieve more accurate and robust SLAM [5], [6], [7].

In the study of the LiDAR-inertial odometry (LIO), the tightly-coupled filter-based LIO systems have received widespread attention [8], [9] due to their better real-time performance and capability to fully leverage the LiDAR and IMU inputs. The efficiency and accuracy are the most important metrics for evaluating their performance. The efforts on the efficiency improvement primarily focus on modifying the spatial data structure of point clouds, optimizing point insertion and query methodologies [10], as well as refining the structure of LiDAR scans [11]. While these methods can significantly enhance efficiency, they contribute minimally to improvements in accuracy. To achieve the better accuracy, researchers focus on the state estimation method of LIO. The most popular estimation methods such as extended Kalman filter (EKF) [8] and iterated EKF [13] primarily reduce estimation errors through two approaches: selecting appropriate expansion points for Taylor expansions and increasing the number of iterations in the estimate update process [12], [13]. The former one mitigates the linearization errors while the latter maximizes the utilization of available measurements to refine the estimation outcomes. Note that the reduction of linearization errors can also make the correction of estimates by measurement information more effective. Thus, this work aims to improve the estimation accuracy by further reducing linearization errors.

To achieve the further reduction of linearization errors in the estimation process, we propose a high-precision tightly-coupled LIO method, named SI-LIO. This method is presented based on the single-iteration invariant extended Kalman filter (IEKF), taking advantage of the right-invariant estimation errors defined on the Lie group. With these estimation errors, we utilize the Lie exponential map between the Lie group and its Lie algebra such that the Taylor expansions only conducted on the orientation estimation errors in state estimation. This reduces the linearization errors, effectively improving the accuracy of the LIO system, particularly in the cases that larger prediction error exists. Moreover, the mitigation of linearization errors enables the state estimation corrections derived from measurement information to be fully realized via a single-iteration update process. This effectively reduces the estimation time in SI-LIO, enhancing its efficiency compared to the FAST-LIO2 with multi-iteration update. The contributions of this work are listed as follows.

- 1) We propose a novel filter-based LIO method, which utilizes the Lie exponential for the linearizations such that the linear approximations are only conducted on the orientation estimation errors in estimation process. This method significantly reduces the linearization errors compared to the state-of-the-art filter-based LIO systems [12], [13].
- 2) The reduction of linearization errors endows SI-LIO with higher accuracy than FAST-LIO2 without requiring multiple iterations in estimate update, especially in the cases with larger prediction errors. Additionally, the employment of the single-iteration scheme reduces the estimation time, thereby enhancing the efficiency of SI-LIO relative to FAST-LIO2.
- 3) We conduct a series of experiments on public datasets, which show the higher accuracy of SI-LIO than the state-of-the-art LIO systems. Specifically, the accuracy of SI-LIO is roughly 15% higher than FAST-LIO2 with single-iteration estimate update. Furthermore, SI-LIO achieves higher precision than FAST-LIO2 with the average system runtime approximately 30% shorter.

## II. RELATED WORKS

Up to now, there have been numerous studies on LIO. In this section, we only introduce the tightly-coupled filter-based LIO systems that are most relevant to our work.

### A. Efficient Filter-Based LiDAR-Inertial Odometry

Ref. [11] focuses on the structure of the LiDAR scan in the filter-based LIO systems, proposing a ring-based fast approximate least squares method for fast normal estimation. Meanwhile, a robust and accurate hierarchical data association method is designed. These efforts improve the efficiency of the LIO effectively. In [10], an incremental voxels-based LIO is presented, which is named Faster-LIO. In this system, the common used tree-based structures for spatial point cloud division are replaced with the incremental voxels, and the corresponding incremental insertion and parallel approximated k-NN queries are applied. This leads to the achievement of trajectory tracking with higher efficiency. Ref. [16] proposes a robocentric voxel map that efficiently manages map points and facilitates k-nearest neighbor queries. The aforementioned efforts on promoting the LIO systems' efficiency focus on the improvement of the LiDAR point cloud data structure and the query strategy, which have a marginal impact on system precision.

### B. High-Precision Filter-Based LiDAR-Inertial Odometry

The most classic filter-based LIO approach is the extended Kalman filter (EKF)-based one. Ref. [8] applies a modified EKF to a point-by-point LIO framework for state estimation. Compared with the traditional EKF, the one used in this LIO achieves accuracy improvement, since it chooses the zero estimation error points rather than the estimates as the Taylor expansion points. Building upon this EKF-based approach, Ref. [12] proposes the FAST-LIO, which utilizes the iterated EKF to calculate more accurate state estimates in fast-motion environment. The iterated EKF reduces the estimation errors via conducting multiple iterations in the estimate update, adequately correcting the estimates by the LiDAR inputs. Based on the FAST-LIO system, the FAST-LIO2 presented in [13] further enhances the accuracy by directly using the raw LiDAR points

for localization and mapping and adopting the incremental k-d tree for map maintaining. Besides, Ref. [9] increases the accuracy by modifying the propagation process of the iterated EKF, using differential and exponential operations to achieve the propagation of estimation error covariance. In summary, the efforts on accuracy improvement applied in the aforementioned filter-based LIO systems primarily includes two perspective, reducing the linearization errors and enhancing the correction of estimates. Inspired by these efforts, to improve the accuracy, we aim to take measures to further reduce linearization errors, which can also make the correction of estimates by measurement information more effective.

## III. PRELIMINARY

The Euclidean rotation group  $\text{SO}(3)$  and transformation group  $\text{SE}(3)$  are two common Lie groups in the research field of robotics. Based on them, a Lie group which is an extension of  $\text{SE}(3)$  is defined as

$$\text{SE}_4(3) = \left\{ \begin{bmatrix} R & \mathbf{a}_1 & \mathbf{a}_2 & \mathbf{a}_3 & \mathbf{a}_4 \\ \mathbf{0}_{4 \times 3} & I_4 & & & \end{bmatrix} \mid R \in \text{SO}(3), \right. \\ \left. \mathbf{a}_1, \mathbf{a}_2, \mathbf{a}_3, \mathbf{a}_4 \in \mathbb{R}^3 \right\}.$$

Its corresponding Lie algebra is denoted as

$$\text{se}_4(3) = \left\{ \begin{bmatrix} \boldsymbol{\omega}^\times & \mathbf{t}_1 & \mathbf{t}_2 & \mathbf{t}_3 & \mathbf{t}_4 \\ \mathbf{0}_{4 \times 3} & \mathbf{0}_{4 \times 4} & & & \end{bmatrix} \mid \boldsymbol{\omega}, \mathbf{t}_1, \mathbf{t}_2, \mathbf{t}_3, \mathbf{t}_4 \in \mathbb{R}^3 \right\},$$

where  $(\cdot)^\times$  represents the skew-symmetric matrices. Specifically, for a vector  $\boldsymbol{\omega} = [\omega_1, \omega_2, \omega_3]^\top \in \mathbb{R}^3$ , its corresponding skew-symmetric matrix is

$$\boldsymbol{\omega}^\times = \begin{bmatrix} 0 & -\omega_3 & \omega_2 \\ \omega_3 & 0 & -\omega_1 \\ -\omega_2 & \omega_1 & 0 \end{bmatrix}.$$

There always exists an exponential map between a Lie group and its corresponding Lie algebra, which is named Lie exponential. For group  $\text{SE}_4(3)$  and its Lie algebra  $\text{se}_4(3)$ , the Lie exponential is the matrix exponential  $\exp(\cdot)$ . There also exists a map  $\mathcal{L}(\cdot) : \mathbb{R}^{15} \rightarrow \text{se}_4(3)$ , which is given as

$$\mathcal{L}(\mathbf{l}) = \begin{bmatrix} \boldsymbol{\omega}^\times & \mathbf{t}_1 & \mathbf{t}_2 & \mathbf{t}_3 & \mathbf{t}_4 \\ \mathbf{0}_{4 \times 3} & \mathbf{0}_{4 \times 4} & & & \end{bmatrix} \in \text{se}_4(3),$$

where  $\mathbf{l} = [\boldsymbol{\omega}^\top, \mathbf{t}_1^\top, \mathbf{t}_2^\top, \mathbf{t}_3^\top, \mathbf{t}_4^\top]^\top \in \mathbb{R}^{15}$ . We denote  $\exp_m(\mathbf{a}) = \exp(\mathcal{L}(\mathbf{a}))$  when  $\mathbf{a}$  represents a vector belonging to  $\mathbb{R}^{15}$  and  $\exp_m(\mathbf{a}) = \exp(\mathbf{a}^\times)$  when  $\mathbf{a} \in \mathbb{R}^3$ .

*Notations:*  $\mathbf{0}$  represents the zero vector or matrix with proper dimension.  $I$  is the identical matrix with proper dimension. Block diagonal matrix  $A$  with matrices  $A_i, i = 1 \dots, n$  on its diagonal is denoted as  $A = \text{diag}(A_i, i = 1 \dots, n)$ .

## IV. SYSTEM MODEL

In this section, we will introduce the overview of SI-LIO, the state transition model, and the LiDAR measurement model.

### A. System Overview

The overview of our LiDAR-inertial odometry (LIO) system, SI-LIO, is presented in Fig. 1. As shown in this figure, the IMU

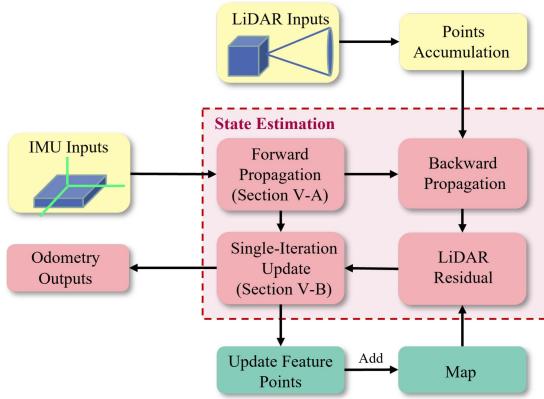


Fig. 1. Overview of SI-LIO.

and LiDAR measurements are used for the state estimation. Once the IMU measurements are obtained, the forward propagation process is performed and the prior state estimate<sup>1</sup> is calculated. In contrast, the LiDAR measurements are accumulated over a period and processed collectively for the estimate update. For these accumulated LiDAR points, the backward propagation process introduced in FAST-LIO [12] is adopted to compensate for motion distortion, and the corresponding residuals are computed. With the residuals and the prior state estimate, the posterior state estimate,<sup>2</sup> which is the odometry output and converges to the one calculated by the maximum a-posteriori (MAP) estimate, can be obtained via the single-iteration estimate update process. Based on the posterior state estimate, these LiDAR points can be registered to the global frame and the map is updated.

### B. State Transition Model

Consider that the IMU is rigidly attached to the LiDAR in our system with the known extrinsic  ${}^I T_L = ({}^I R_L, {}^I p_L)$  from LiDAR to IMU. Take the IMU frame as the body frame and the first IMU frame as the world frame for convenience, and the kinematic model of the system is derived as

$$\begin{aligned} {}^W \dot{R} &= {}^W R (\omega_m - b_\omega - n_\omega)^\times, \quad {}^W \dot{p} = {}^W v, \\ {}^W \dot{v} &= {}^W R (a_m - b_a - n_a) + {}^W g, \quad \dot{b}_\omega = n_{bw}, \quad \dot{b}_a = n_{ba}. \end{aligned} \quad (1)$$

The left superscript  $W$  means that the vector is in the world frame.  ${}^W R \in \text{SO}(3)$  and  ${}^W p \in \mathbb{R}^3$  represent the orientation and position, which form the pose.  ${}^W v$  denotes the velocity.  $\omega_m$  and  $a_m$  are the measurements collected by the gyroscope and accelerometer of the IMU with the biases  $b_\omega$  and  $b_a$ , and Gaussian measurement noises  $n_\omega \sim N(\mathbf{0}, \sigma_\omega^2 I)$  and  $n_a \sim N(\mathbf{0}, \sigma_a^2 I)$ . The biases are modeled as random walk processes with Gaussian noises  $n_{bw} \sim N(\mathbf{0}, \sigma_{bw}^2 I)$  and  $n_{ba} \sim N(\mathbf{0}, \sigma_{ba}^2 I)$ . The gravitational acceleration  ${}^W g$  is constant. This model is a common form used in most of the researches on LIO [12], [13], [17].

With  $i \in \{0, 1, 2, \dots\}$  representing the index of sampling time instants of IMU, the continuous-time kinematics (1) can

<sup>1</sup>The estimate calculated by each forward propagation process before the update using the LiDAR inputs.

<sup>2</sup>The estimate obtained after each update using the LiDAR residuals.

be discretized as

$$\begin{aligned} {}^W R_{i+1} &= {}^W R_i \exp_m ((\omega_{m,i} - b_{\omega,i} - n_{\omega,i}) \Delta t), \\ {}^W p_{i+1} &= {}^W p_i + {}^W v_i \Delta t, \\ {}^W v_{i+1} &= {}^W v_i + [{}^W R_i (a_{m,i} - b_{a,i} - n_{a,i}) + {}^W g] \Delta t, \\ b_{\omega,i+1} &= b_{\omega,i} + n_{bw,i} \Delta t, \quad b_{a,i+1} = b_{a,i} + n_{ba,i} \Delta t, \end{aligned} \quad (2)$$

where  $\Delta t$  represents the sampling period of IMU. Motivated by [18], we define the system state as

$$X = \left[ \begin{array}{c|ccccc} {}^W R & {}^W p & {}^W v & b_\omega & b_a \\ \hline \mathbf{0}_{4 \times 3} & & & \bar{I}_4 & \end{array} \right] \quad (3)$$

such that the matrix Lie group  $\text{SE}_4(3)$  becomes the state space. Then, the discrete-time kinematic model (2) can be written into the compact form

$$X_{i+1} = f(X_i, u_i, n_i)$$

with the input  $u_i = [\omega_{m,i}^\top, a_{m,i}^\top]^\top$  and the noise  $n_i = [n_{\omega,i}^\top, n_{a,i}^\top, n_{bw,i}^\top, n_{ba,i}^\top]^\top$ .

As given in (3), we incorporate the IMU biases along with the system pose and velocity to form the system state, which is motivated by [13]. Since the evolution of the system pose is related to the biases and the biases exhibit randomness, considering them as a part of the system state to be estimated can improve the estimation accuracy of the system pose. Furthermore, when involving the biases in the state, compared to the state space  $\text{SO}(3) \times \mathbb{R}^n$  used in [12], [13], choosing the matrix Lie group  $\text{SE}_4(3)$  as the state space can reduce the linearization errors in the estimation process. This is attributable to the fact that the invariant extended Kalman filter (IEKF), which is capable of fundamentally reducing linearization errors, is applicable for  $\text{SE}_4(3)$  rather than  $\text{SO}(3) \times \mathbb{R}^n$ . The details for the reduction of the linearization errors in IEKF are illustrated in Section V.

### C. LiDAR Measurement Model

The LiDAR measurements are coordinates of points in the LiDAR frame. We always accumulate them over a certain period of time, which is called a LiDAR scan, and then process them collectively. We use  $k \in \{1, 2, 3, \dots\}$  to denote the index of each LiDAR scan's ending time, referring it as time  $k$  for convenience hereafter.  $k = 0$  corresponds to the initial time. At time  $k$ , accumulated LiDAR points are used for the estimate update. It is known that coordinates of these points are not in the same frame since they are not collected at the same time. Therefore, we perform the backward propagation [12], projecting all these coordinates to the LiDAR frame at time  $k$ . Suppose that at time  $k$ , there are  $M_k$  LiDAR points, the coordinate of the  $j$ th point after the projection satisfies

$${}^L_k p_{f_j} = {}^W R_{L_k}^\top \left( {}^W p_{f_j} - {}^W p_{L_k} \right) + n_k^L. \quad (4)$$

Herein,  ${}^W R_{L_k}$  and  ${}^W p_{L_k}$  are the orientation and position of the LiDAR in the world frame at time  $k$ .  ${}^W p_{f_j}$  denotes the position of the point in the world frame.  $n_k^L$  is the Gaussian measurement noise of LiDAR, satisfying  $n_k^L \sim N(\mathbf{0}, \sigma_L^2 I)$  for any  $k \geq 0$ .

For each LiDAR point, we consider that it belongs to the closest plane defined by the nearby map points, i.e.,

$$0 = (\mu_j)^\top ({}^W p_{f_j} - q_j) \quad (5)$$

for the  $j$ th,  $j = 1 \dots, M_k$  LiDAR point at time  $k$ . Herein,  $\mu_j$  represents the unit normal vector of the plane that the point belongs to and  $q_j$  is the position of a point on the plane. Based on (5), we define

$$\begin{aligned} h_{j,k}(X_k, \mathbf{n}_k^L) \\ = (\boldsymbol{\mu}_j)^\top \left[ {}^W R_k {}^I R_L \left( {}^{L_k} \mathbf{p}_{f_j} - \mathbf{n}_k^L \right) + {}^W R_k {}^I \mathbf{p}_L + {}^W \mathbf{p}_k - q_j \right] = 0, \end{aligned} \quad (6)$$

where  $X_k$  is the state at time  $k$  with  ${}^W R_k$  and  ${}^W \mathbf{p}_k$  being the orientation and position, respectively.  $h_{j,k}$  is the residual function.

## V. STATE ESTIMATION

IEKF is a stable observer for the systems defined on matrix Lie groups [14]. We adopt an IEKF-based estimation method to estimate the system state defined in (3). The estimation process includes two key steps: forward propagation and single-iteration estimate update. The former one occurs once the IMU inputs are obtained, while the latter happens at the end of LiDAR scans. Since the sampling period  $\Delta t$  of IMU is significantly shorter than the duration of each LiDAR scan, multiple forward propagation processes are performed during a LiDAR scan.

### A. Forward Propagation

Denote the posterior estimate obtained via the latest update step as  $\hat{X}_{k-1}$  and the corresponding error covariance estimate as  $\hat{P}_{k-1}$ . Then, based on them, the propagation step is performed whenever the IMU inputs are received before the next update.

In the forward propagation, the state estimate evolves according to (2) with the process noise being set to zero:

$$\hat{X}_{i+1} = f(\hat{X}_i, \mathbf{u}_i, \mathbf{0}), \quad \hat{X}_0 = \bar{X}_{k-1}. \quad (7)$$

The state estimates

$$\hat{X}_i = \begin{bmatrix} {}^W \hat{R}_i & {}^W \hat{\mathbf{p}}_i & {}^W \hat{\mathbf{v}}_i & \hat{\mathbf{b}}_{\omega,i} & \hat{\mathbf{b}}_{a,i} \\ \mathbf{0}_{4 \times 3} & & I_4 & & \end{bmatrix},$$

$$\bar{X}_{k-1} = \begin{bmatrix} {}^W \bar{R}_{k-1} & {}^W \bar{\mathbf{p}}_{k-1} & {}^W \bar{\mathbf{v}}_{k-1} & \bar{\mathbf{b}}_{\omega,k-1} & \bar{\mathbf{b}}_{a,k-1} \\ \mathbf{0}_{4 \times 3} & & I_4 & & \end{bmatrix}.$$

To propagate the corresponding error covariance estimate, we define the estimation errors in the group  $\text{SE}_4(3)$ [14] as

$$\hat{\eta}_i = X_i \hat{X}_i^{-1}, \quad \bar{\eta}_{k-1} = X_{k-1} \bar{X}_{k-1}^{-1}. \quad (8)$$

These errors are right-invariant, i.e., they are invariant to right multiplications which refer to the transformations of the form  $X_i \rightarrow X_i X_0$ ,  $\hat{X}_i \rightarrow \hat{X}_i X_0$ [18]. The matrix  $X_0 \in \text{SE}_4(3)$ . This is the reason that the observer is named invariant extended Kalman filter. With the Lie exponential  $\exp(\cdot)$  and the map  $\mathcal{L}(\cdot) : \mathbb{R}^{15} \rightarrow \text{se}_4(3)$ , there must exist error vectors

$$\hat{\xi}_i = \begin{bmatrix} \hat{\boldsymbol{\omega}}_i \\ {}^W \mathbf{p}_i - {}^W R_i {}^W \hat{R}_i^\top \times {}^W \hat{\mathbf{p}}_i \\ {}^W \mathbf{v}_i - {}^W R_i {}^W \hat{R}_i^\top \times {}^W \hat{\mathbf{v}}_i \\ \mathbf{b}_{\omega,i} - {}^W R_i {}^W \hat{R}_i^\top \times \hat{\mathbf{b}}_{\omega,i} \\ \mathbf{b}_{a,i} - {}^W R_i {}^W \hat{R}_i^\top \times \hat{\mathbf{b}}_{a,i} \end{bmatrix}$$

$$\bar{\xi}_k = \begin{bmatrix} \bar{\boldsymbol{\omega}}_k \\ {}^W \mathbf{p}_k - {}^W R_k {}^W \bar{R}_k^\top \times {}^W \bar{\mathbf{p}}_k \\ {}^W \mathbf{v}_k - {}^W R_k {}^W \bar{R}_k^\top \times {}^W \bar{\mathbf{v}}_k \\ \mathbf{b}_{\omega,k} - {}^W R_k {}^W \bar{R}_k^\top \times \bar{\mathbf{b}}_{\omega,k} \\ \mathbf{b}_{a,k} - {}^W R_k {}^W \bar{R}_k^\top \times \bar{\mathbf{b}}_{a,k} \end{bmatrix},$$

such that

$$\hat{\eta}_i = \exp_m(\hat{\xi}_i), \quad \bar{\eta}_{k-1} = \exp_m(\bar{\xi}_{k-1}).$$

Herein, the orientation estimation errors  ${}^W R_i {}^W \hat{R}_i^\top = \exp_m(\hat{\boldsymbol{\omega}}_i)$  and  ${}^W R_{k-1} {}^W \bar{R}_{k-1}^\top = \exp_m(\bar{\boldsymbol{\omega}}_{k-1})$ . Note that there is no linearization error introduced in the mapping process from  $\hat{\eta}_i, \bar{\eta}_{k-1}$  to  $\hat{\xi}_i, \bar{\xi}_{k-1}$ . By taking the Taylor expansions of these orientation estimation errors at  $\hat{\boldsymbol{\omega}}_i = \mathbf{0}$  and  $\bar{\boldsymbol{\omega}}_{k-1} = \mathbf{0}$ , the errors can be approximated as

$$\hat{\xi}_i = \begin{bmatrix} \hat{\boldsymbol{\omega}}_i \\ {}^W \hat{\mathbf{p}}_i - \hat{\boldsymbol{\omega}}_i \times {}^W \hat{\mathbf{p}}_i \\ {}^W \hat{\mathbf{v}}_i - \hat{\boldsymbol{\omega}}_i \times {}^W \hat{\mathbf{v}}_i \\ \hat{\mathbf{b}}_{\omega,i} - \hat{\mathbf{b}}_{\omega,i} - \hat{\boldsymbol{\omega}}_i \times \hat{\mathbf{b}}_{\omega,i} \\ \hat{\mathbf{b}}_{a,i} - \hat{\mathbf{b}}_{a,i} - \hat{\boldsymbol{\omega}}_i \times \hat{\mathbf{b}}_{a,i} \end{bmatrix},$$

$$\bar{\xi}_k = \begin{bmatrix} \bar{\boldsymbol{\omega}}_k \\ {}^W \bar{\mathbf{p}}_k - \bar{\boldsymbol{\omega}}_k \times {}^W \bar{\mathbf{p}}_k \\ {}^W \bar{\mathbf{v}}_k - \bar{\boldsymbol{\omega}}_k \times {}^W \bar{\mathbf{v}}_k \\ \bar{\mathbf{b}}_{\omega,k} - \bar{\mathbf{b}}_{\omega,k} - \bar{\boldsymbol{\omega}}_k \times \bar{\mathbf{b}}_{\omega,k} \\ \bar{\mathbf{b}}_{a,k} - \bar{\mathbf{b}}_{a,k} - \bar{\boldsymbol{\omega}}_k \times \bar{\mathbf{b}}_{a,k} \end{bmatrix}, \quad (9)$$

and the linear error evolution model

$$\hat{\xi}_{i+1} = A_i \hat{\xi}_i + \Delta t B_i \mathbf{n}_i, \quad \hat{\xi}_0 = \bar{\xi}_{k-1} \quad (10)$$

is obtained based on (2) and (7). In this model,

$$\begin{aligned} A_i &= \begin{bmatrix} A_{i,11} & \mathbf{0} & \mathbf{0} & -\Delta t {}^W \hat{R}_i & \mathbf{0} \\ A_{i,21} & I & \Delta t I & A_{i,24} & \mathbf{0} \\ A_{i,31} & \mathbf{0} & I & A_{i,34} & -\Delta t {}^W \hat{R}_i \\ A_{i,41} & \mathbf{0} & \mathbf{0} & A_{i,44} & \mathbf{0} \\ A_{i,51} & \mathbf{0} & \mathbf{0} & -\Delta t (\hat{\mathbf{b}}_{a,i})^\times {}^W \hat{R}_i & I \end{bmatrix}, \\ B_i &= \begin{bmatrix} -{}^W \hat{R}_i & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ -({}^W \hat{\mathbf{p}}_i + {}^W \hat{\mathbf{v}}_i \Delta t)^\times \hat{R}_i & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \hat{B}_i & -{}^W \hat{R}_i & \mathbf{0} & \mathbf{0} \\ -(\hat{\mathbf{b}}_{\omega,i})^\times {}^W \hat{R}_i & \mathbf{0} & I & \mathbf{0} \\ -(\hat{\mathbf{b}}_{a,i})^\times {}^W \hat{R}_i & \mathbf{0} & \mathbf{0} & I \end{bmatrix}. \end{aligned}$$

Herein, matrices

$$A_{i,11} = I + {}^W \hat{R}_i (\hat{\mathbf{b}}_{\omega,i})^\times,$$

$$A_{i,21} = ({}^W \hat{\mathbf{p}}_i + \Delta t {}^W \hat{\mathbf{v}}_i)^\times {}^W \hat{R}_i (\hat{\mathbf{b}}_{\omega,i})^\times,$$

$$A_{i,24} = -\Delta t ({}^W \hat{\mathbf{p}}_i + \Delta t {}^W \hat{\mathbf{v}}_i)^\times {}^W \hat{R}_i,$$

$$\begin{aligned}
A_{i,31} &= \left[ {}^W \hat{\mathbf{v}}_i + \left( {}^W \hat{R}_i \left( \mathbf{a}_{m,i} - \hat{\mathbf{b}}_{a,i} \right) \right. \right. \\
&\quad \left. \left. + {}^W \mathbf{g} \right) \Delta t \right]^{{}^W \hat{R}_i \left( \hat{\mathbf{b}}_{\omega,i} \right)} \\
&\quad + \left[ {}^W \hat{R}_i \left( \mathbf{a}_{m,i} - \hat{\mathbf{b}}_{a,i} \right) \right]^{{}^W \hat{R}_i \left( \mathbf{a}_{m,i} \right)} \Delta t - \Delta t \left( {}^W \hat{R}_i \mathbf{a}_{m,i} \right)^{{}^W \hat{R}_i} \\
&\quad + \Delta t {}^W \hat{R}_i \hat{\mathbf{b}}_{a,i}^{{}^W \hat{R}_i}, \\
A_{i,34} &= -\Delta t \left[ {}^W \hat{\mathbf{v}}_i + \left( {}^W \hat{R}_i \left( \mathbf{a}_{m,i} - \hat{\mathbf{b}}_{a,i} \right) + {}^W \mathbf{g} \right) \Delta t \right]^{{}^W \hat{R}_i}, \\
A_{i,41} &= \left( \hat{\mathbf{b}}_{\omega,i} \right)^{{}^W \hat{R}_i} \left( \hat{\mathbf{b}}_{\omega,i} \right)^{{}^W \hat{R}_i}, \\
A_{i,44} &= I - \Delta t \left( \hat{\mathbf{b}}_{\omega,i} \right)^{{}^W \hat{R}_i} {}^W \hat{R}_i, \\
A_{i,51} &= \left( \hat{\mathbf{b}}_{a,i} \right)^{{}^W \hat{R}_i} \left( \hat{\mathbf{b}}_{\omega,i} \right)^{{}^W \hat{R}_i}, \\
\hat{B}_i &= -\left[ {}^W \hat{\mathbf{v}}_i + \left( {}^W \hat{R}_i \left( \mathbf{a}_{m,i} - \hat{\mathbf{b}}_{a,i} \right) + {}^W \mathbf{g} \right) \Delta t \right]^{{}^W \hat{R}_i}.
\end{aligned}$$

With the linearized evolution rule of the estimation error (10), the propagation of the error covariance estimate satisfies

$$\hat{P}_{i+1} = A_i \hat{P}_i A_i^\top + \Delta t^2 B_i Q B_i^\top, \quad \hat{P}_0 = \bar{P}_{k-1}.$$

$Q$  is the covariance matrix of the noise vector  $\mathbf{n}_i$ , which satisfies  $Q = \text{diag}(\sigma_\omega^2 I, \sigma_a^2 I, \sigma_{ba}^2 I, \sigma_{ba}^2 I)$ .

### B. Single-Iteration Update

At the end of each LiDAR scan, we calculate the residuals corresponding to the accumulated LiDAR points and utilize them to update the state estimate, obtaining the odometry output.

At time  $k$ ,  $k = 1, 2, \dots$ , denote the prior state estimate calculated by the latest forward propagation as

$$\hat{X}_k = \begin{bmatrix} {}^W \hat{R}_k & {}^W \hat{\mathbf{p}}_k & {}^W \hat{\mathbf{v}}_k & \hat{\mathbf{b}}_{\omega,k} & \hat{\mathbf{b}}_{a,k} \\ \mathbf{0}_{4 \times 3} & & & & \bar{I}_4 \end{bmatrix}$$

and the corresponding error covariance estimate as  $\hat{P}_k$ . Then, utilizing the point coordinate given in (4), we can calculate the residual corresponding to the  $j$ th,  $j = 1, 2, \dots, M_k$  LiDAR point as

$$\begin{aligned}
h_{j,k}(\hat{X}_k, \mathbf{0}) &= \\
(\boldsymbol{\mu}_j)^\top \left[ {}^W \hat{R}_k {}^I R_L {}^{L_k} \mathbf{p}_{f_j} + {}^W \hat{R}_k {}^I \mathbf{p}_L + {}^W \hat{\mathbf{p}}_k - \mathbf{q}_j \right]. \quad (11)
\end{aligned}$$

It can be linearized as

$$\begin{aligned}
h_{j,k}(X_k, \mathbf{n}_k^L) - h_{j,k}(\hat{X}_k, \mathbf{0}) &= \\
= \mathbf{0} - h_{j,k}(\hat{X}_k, \mathbf{0}) &= H_{j,k} \hat{\boldsymbol{\xi}}_k + n_k^{j,L} \quad (12)
\end{aligned}$$

with the error vector

$$\hat{\boldsymbol{\xi}}_k = \begin{bmatrix} \hat{\omega}_k \\ {}^W \mathbf{p}_k - {}^W \hat{\mathbf{p}}_k - \hat{\omega}_k \times {}^W \hat{\mathbf{p}}_k \\ {}^W \mathbf{v}_k - {}^W \hat{\mathbf{v}}_k - \hat{\omega}_k \times {}^W \hat{\mathbf{v}}_k \\ \hat{\mathbf{b}}_{\omega,k} - \hat{\mathbf{b}}_{\omega,k} - \hat{\omega}_k \times \hat{\mathbf{b}}_{\omega,k} \\ \hat{\mathbf{b}}_{a,k} - \hat{\mathbf{b}}_{a,k} - \hat{\omega}_k \times \hat{\mathbf{b}}_{a,k} \end{bmatrix}. \quad (13)$$

In (12), the measurement matrix

$$H_{j,k} = [\tilde{H}_{j,k} \quad \boldsymbol{\mu}_j^\top \quad \mathbf{0} \quad \mathbf{0} \quad \mathbf{0}]$$

and the Gaussian noise satisfying  $n_k^{j,L} \sim N(\mathbf{0}, \sigma_L^2)$ . Herein,  $\tilde{H}_{j,k} = -\boldsymbol{\mu}_j^\top ({}^W \hat{R}_k ({}^I R_L {}^{L_k} \mathbf{p}_{f_j} + {}^I \mathbf{p}_L) + {}^W \hat{\mathbf{p}}_k)$ . The specific linearization process of the residual (11) is given in Appendix. Note that the linearization is not achieved through the Taylor expansion of the residual function, but rather via a direct transformation using the error vector (9). In (14), only an approximation of  ${}^W R_k {}^W \hat{R}_k^\top = \exp_m(\hat{\omega}_k)$  is conducted to obtain the third equation, which refers to the Taylor expansion of  $\exp_m(\hat{\omega}_k)$  at  $\hat{\omega}_k = \mathbf{0}$ .

The residuals corresponding to all the  $M_k$  points comprise the vector  $\mathbf{h}_k(\hat{X}_k, \mathbf{0}) = [h_{1,k}(\hat{X}_k, \mathbf{0}), \dots, h_{M_k,k}(\hat{X}_k, \mathbf{0})]^\top$ , then it is obtained that

$$h_k(X_k, \mathbf{n}_k^L) - h_k(\hat{X}_k, \mathbf{0}) = \mathbf{0} - h_k(\hat{X}_k, \mathbf{0}) = H_k \hat{\boldsymbol{\xi}}_k + \mathbf{n}_k^L,$$

where  $H_k = [H_{1,k}^\top, \dots, H_{M_k,k}^\top]^\top$  and noise  $\mathbf{n}_k^L \sim N(\mathbf{0}, \sigma_L^2 I)$ . Using these residuals, the state estimate and the error covariance estimate can be updated according to

$$\begin{aligned}
\bar{X}_k &= \exp_m(-K_k h_k(\hat{X}_k, \mathbf{0})) \hat{X}_k, \\
\bar{P}_k &= \hat{P}_k - K_k H_k \hat{P}_k
\end{aligned}$$

with the Kalman gain [12]

$$K_k = \frac{1}{\sigma_L^2} \left( \frac{1}{\sigma_L^2} H_k^\top H_k + \hat{P}_k^{-1} \right)^{-1} H_k^\top.$$

Throughout the aforementioned estimation process, linear approximations are exclusively applied to orientation estimation errors, which pertain to a single component, orientation, within the system state. These approximations refer to the Taylor expansions of  ${}^W R_k {}^W \hat{R}_k^\top = \exp_m(\hat{\omega})$  at  $\hat{\omega} = \mathbf{0}$ , and are conducted during the derivation of the linear error evolution model (10) and the linearization of the residual function (11). Compared to traditional EKF and the iterated EKF [12], where Taylor expansions are used for system kinematics, nonlinear error evolution model, and residual functions, the IEKF-based method presented in this section effectively reduces the linearization errors, enhancing the estimation accuracy thereby.

## VI. EXPERIMENTS

To demonstrate the performance of our SI-LIO, we conduct a series of experiments on public datasets, comparing it with the existing state-of-the-art LIO systems to highlight its advantages.<sup>3</sup> The LIO systems involved in the comparison include FAST-LIO2 [13], Point-LIO [8], and LIO-SAM [21]. The former two are popular filter-based LIO systems, while the latter one is optimization-based. Note that our SI-LIO is basically developed under FAST-LIO2, i.e., the extraction and preprocessing of the environment information, the spacial data structure of point cloud, and the query strategy of SI-LIO are identical to those of FAST-LIO2. The frequency of the estimate update is set to 5Hz for both SI-LIO and FAST-LIO2 in the experiments. At this frequency, the prediction errors of IMU are greater than at 10 Hz. The datasets we choose for the experiments include UTBM

<sup>3</sup>The experimental video is available at <https://youtu.be/vwxg8KNcwgI>.

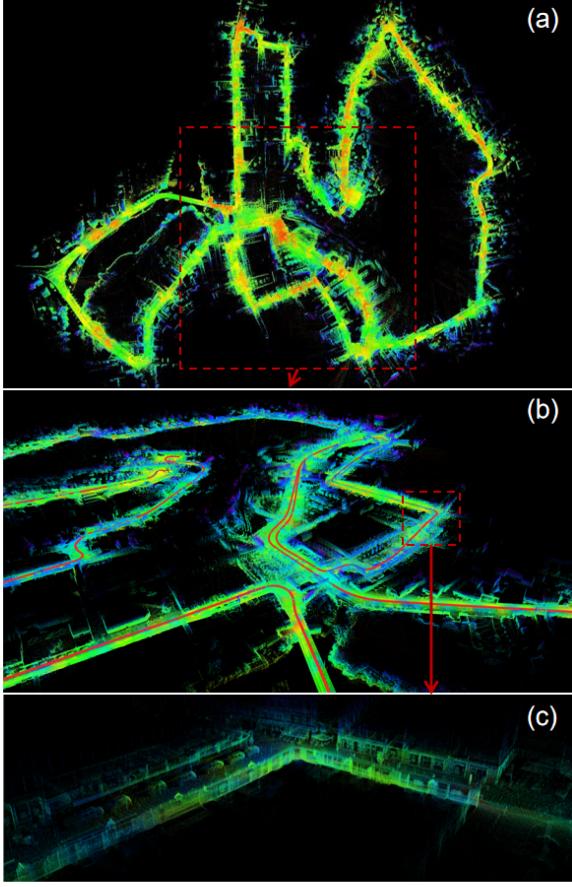


Fig. 2. (a) Localization and mapping result of SI-LIO on a sequence of 5.03 km chosen from UTBM robocar dataset [19]. (b) The zoomed-in view image of the red box in (a), which presents the details of the localization result. (c) The zoomed-in view image of the red box in (b), which presents the details of the mapping result.

robocar dataset [19] and M2DGR [20], which can provide both IMU and LiDAR inputs. The sequences in them correspond to paths of varying shapes and scales and are valid for all the odometry systems under evaluation.

#### A. Effectiveness Experiments

To demonstrate the effectiveness of SI-LIO, we select one sequence from UTBM and run SI-LIO on it. The length of the sequence is 5.03 km. The corresponding result of localization and mapping is presented in Fig. 2, which demonstrates the achievement of the accurate simultaneous localization and mapping by SI-LIO.

#### B. Accuracy Experiments

We compare the accuracy of our SI-LIO against FAST-LIO2, LIO-SAM, and Point-LIO by running them on the sequences chosen from dataset M2DGR and calculating the corresponding absolute translational errors (ATE, RMSE). The dataset M2DGR has high-precision ground truth of trajectories collected by RTK sensors, and the sequences we select are in different lengths and shapes, which guarantees the fair comparison.

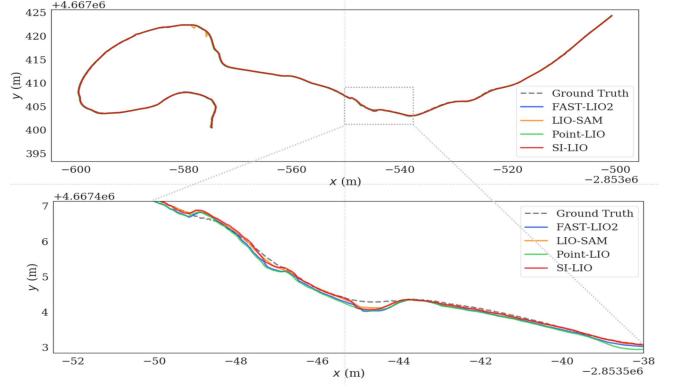


Fig. 3. Comparison among the ground truth and the trajectory estimates obtained by SI-LIO, FAST-LIO2, LIO-SAM, and Point-LIO when run on the sequence m2dgr-gate\_03.

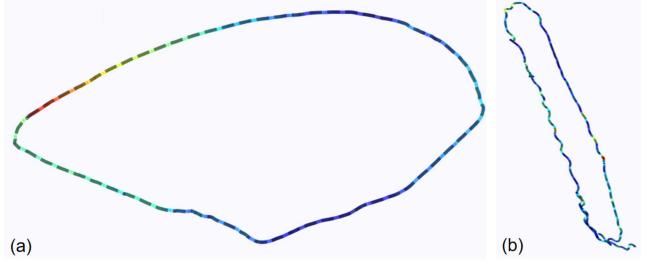


Fig. 4. Trajectory estimates calculated by SI-LIO when run on the sequences m2dgr-street\_04 (a) and m2dgr-street\_08 (b), and the corresponding ground truth. The dashed lines represent the ground truth, while the colored solid lines depict trajectory estimates. The color indicates the magnitude of estimation errors: red-large, blue-small.

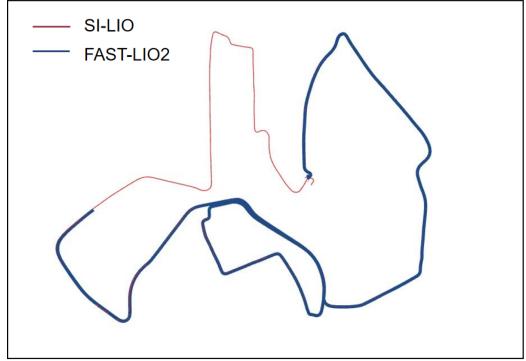


Fig. 5. Trajectory estimates of the sequence shown by Fig. 2. The estimates are obtained by SI-LIO and FAST-LIO2 when SI-LIO completes the estimation of the entire motion trajectory. When SI-LIO completes the trajectory estimation, FAST-LIO2 does not.

Since the single-iteration estimate update is applied in SI-LIO, we also set the number of iteration in estimate update to be one for FAST-LIO2. Table I shows that in this case, the ATEs of SI-LIO are approximately 15% smaller than those introduced by FAST-LIO2. That is to say, SI-LIO achieves higher accuracy than FAST-LIO2 when they both iterate once in the estimate update process. As the number of the iteration increases, the accuracy of FAST-LIO2 is enhanced which is shown in Table I. However, the improvement is limited and the accuracy of FAST-LIO2 with multi-iteration estimate update still cannot surpass

TABLE I  
ACCURACY EVALUATION IN ATE (METERS)

MAP ID	FAST-LIO2	FAST-LIO2-SI	SI-LIO	SI-LIO-MI	LIO-SAM	Point-LIO	Dist(km)
m2dgr-street_02	2.88	2.94	<b>2.35</b>	<b>2.35</b>	4.31	3.44	1.48
m2dgr-street_03	0.18	0.19	<b>0.14</b>	<b>0.14</b>	-	0.19	0.42
m2dgr-street_04	<b>0.45</b>	0.47	0.48	0.48	0.96	0.52	0.84
m2dgr-street_05	0.36	0.43	0.34	<b>0.33</b>	0.41	0.46	0.42
m2dgr-street_07	2.92	3.05	2.29	2.28	-	<b>1.53</b>	0.64
m2dgr-street_08	0.23	0.23	<b>0.17</b>	<b>0.17</b>	-	0.23	0.34
m2dgr-gate_01	0.18	0.19	<b>0.14</b>	0.15	0.39	0.19	0.14
m2dgr-gate_02	0.33	0.33	<b>0.28</b>	<b>0.28</b>	0.35	0.34	0.29
m2dgr-gate_03	0.22	0.22	<b>0.12</b>	<b>0.12</b>	0.13	0.24	0.25

<sup>1</sup> Notation “-” means that the LIO system fails on this sequence because of large drift.

<sup>2</sup> FAST-LIO2-SI: FAST-LIO2 with single iteration in the estimate update process.

<sup>3</sup> SI-LIO-MI: SI-LIO with the estimate update process including multiple iterations.

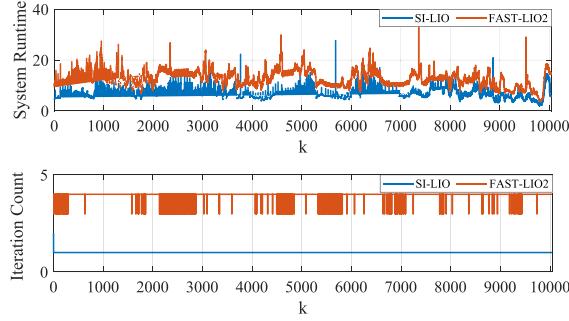


Fig. 6. Iteration count and system runtime corresponding to each estimate update, when SI-LIO and FAST-LIO2 are run on the sequence shown by Fig. 2.

that of SI-LIO. Thus, it is concluded that SI-LIO can achieve higher accuracy than FAST-LIO2 without requiring the multiple iterations in estimate update. Moreover, Table I presents that the precision of SI-LIO is hardly affected by the number of iteration in the estimate update. This indicates that the corrective effects of the measurement information on the estimates can be fully realized through a single iteration in SI-LIO, eliminating the need for multiple iterations.

Besides FAST-LIO2, the accuracy of SI-LIO is also compared with another filter-based LIO, Point-LIO, which utilizes the EKF for the state estimation. As shown in Table I, the accuracy of SI-LIO is also higher than that of Point-LIO, indicating that the high-frequency estimate update in Point-LIO cannot compensate the impact of linearization errors on the system accuracy. Furthermore, from the comparison experiments, we obtain that SI-LIO can achieve higher precision than LIO-SAM, an advanced optimization-based LIO. This further confirms the high precision of SI-LIO. Figs. 3 and 4 show the comparison between the trajectory estimates calculated by various LIO systems, as well as that between the trajectory estimates obtained by SI-LIO and the ground truth.

### C. Efficiency Experiments

From the accuracy comparison experiments, we know that compared with FAST-LIO2, SI-LIO can achieve higher accuracy with only single iteration in estimate update. The reduction in the number of iteration effectively shortens the estimation time. As shown in Table II, the average system runtime of SI-LIO is about 30% shorter than that of FAST-LIO2. The system runtime refers to the time required for the LIO system to complete one cycle

TABLE II  
TIME EVALUATION IN AVERAGE SYSTEM RUNTIME (MILLISECONDS)

MAP ID	FAST-LIO2	SI-LIO
m2dgr-street_02	16.59	<b>10.45</b>
m2dgr-street_03	20.54	<b>13.56</b>
m2dgr-street_04	17.21	<b>11.45</b>
m2dgr-street_05	14.79	<b>10.42</b>
m2dgr-street_07	21.19	<b>13.06</b>
m2dgr-street_08	20.51	<b>14.17</b>
m2dgr-gate_01	21.74	<b>14.15</b>
m2dgr-gate_02	19.14	<b>11.42</b>
m2dgr-gate_03	17.69	<b>12.19</b>

of state estimation and map update. Figs. 5 and 6 intuitively illustrate the higher efficiency of SI-LIO compared to FAST-LIO2 and the reason behind for achieving it.

## VII. CONCLUSION

This letter has presented a single-iteration IEKF-based LIO, named SI-LIO, in which tightly-coupled fusion of the LiDAR and IMU measurements is conducted by an IEKF-based method. This odometry can achieve higher accuracy than the state-of-the-art filter-based LIO systems, since the utilization of Lie exponential fundamentally reduces the linearization errors in the estimation process. Moreover, since IEKF-based estimation method used in SI-LIO can achieve the higher precision without requiring the multiple iterations in the estimate update process, its efficiency is higher than FAST-LIO2. A series of experiments have been conducted on public datasets to demonstrate the performance of SI-LIO. The experiment results show that SI-LIO can achieve higher accuracy than the state-of-the-art LIO systems. Specifically, when the estimate update of FAST-LIO2 also contains a single iteration, the accuracy of SI-LIO is 15% higher than that of FAST-LIO2.

## APPENDIX

In this section, we show the linearization process of the residual corresponding to each LiDAR point. From (6) and (11), it is known that

$$\begin{aligned}
 & h_{j,k}(X_k, \mathbf{n}_k^L) - h_{j,k}(\hat{X}_k, \mathbf{0}) \\
 &= (\boldsymbol{\mu}_j)^\top \left[ \left( {}^W R_k - {}^W \hat{R}_k \right) \left( {}^I R_L {}^{L_k} \mathbf{p}_{f_j} + {}^I \mathbf{p}_L \right) \right. \\
 &\quad \left. + {}^W \mathbf{p}_k - {}^W \hat{\mathbf{p}}_k \right] + (\boldsymbol{\mu}_j)^\top {}^W R_k {}^I R_L \mathbf{n}_k^L.
 \end{aligned}$$

For the terms in the right-hand side without noise, we have

$$\begin{aligned}
 & (\boldsymbol{\mu}_j)^\top \left[ \left( {}^W R_k - {}^W \hat{R}_k \right) \left( {}^I R_L {}^{L_k} \mathbf{p}_{f_j} + {}^I \mathbf{p}_L \right) + {}^W \mathbf{p}_k - {}^W \hat{\mathbf{p}}_k \right] \\
 &= (\boldsymbol{\mu}_j)^\top \left[ \left( {}^W R_k {}^W \hat{R}_k^\top - I \right) {}^W \hat{R}_k \left( {}^I R_L {}^{L_k} \mathbf{p}_{f_j} + {}^I \mathbf{p}_L \right) \right. \\
 &\quad \left. + {}^W \mathbf{p}_k - {}^W \hat{\mathbf{p}}_k - \hat{\omega}_k \times {}^W \hat{\mathbf{p}}_k + \hat{\omega}_k \times {}^W \hat{\mathbf{p}}_k \right] \\
 &= (\boldsymbol{\mu}_j)^\top \left[ \hat{\omega}_k \times \left[ {}^W \hat{R}_k \left( {}^I R_L {}^{L_k} \mathbf{p}_{f_j} + {}^I \mathbf{p}_L \right) + {}^W \hat{\mathbf{p}}_k \right] \right. \\
 &\quad \left. + {}^W \mathbf{p}_k - {}^W \hat{\mathbf{p}}_k - \hat{\omega}_k \times {}^W \hat{\mathbf{p}}_k \right]. \tag{14}
 \end{aligned}$$

Then, according to (13), the measurement matrix corresponding to the  $j$ th,  $j = 1, \dots, M_k$  point can be obtained as

$$H_{j,k} = \begin{bmatrix} \tilde{H}_{j,k} & \boldsymbol{\mu}_j^\top & \mathbf{0} & \mathbf{0} & \mathbf{0} \end{bmatrix}$$

with

$$\tilde{H}_{j,k} = -\boldsymbol{\mu}_j^\top \left( {}^W \hat{R}_k \left( {}^I R_L {}^{L_k} \mathbf{p}_{f_j} + {}^I \mathbf{p}_L \right) + {}^W \hat{\mathbf{p}}_k \right)^\times.$$

As for the noise term

$$n_k^{j,L} = (\boldsymbol{\mu}_j)^\top {}^W R_k {}^I R_L \mathbf{n}_k^L,$$

one has

$$\begin{aligned}
 E \left[ n_k^{j,L} \left( n_k^{j,L} \right)^\top \right] &= E \left[ (\boldsymbol{\mu}_j)^\top {}^W R_k {}^I R_L \mathbf{n}_k^L \mathbf{n}_k^{L^\top} {}^I R_L^\top {}^W R_k^\top \boldsymbol{\mu}_j \right] \\
 &= \sigma_L^2 \boldsymbol{\mu}_j^\top \boldsymbol{\mu}_j = \sigma_L^2.
 \end{aligned}$$

Thus, the linearization result of the residual can be written in the form

$$-h_{j,k}(\hat{X}_k, \mathbf{0}) = H_{j,k} \hat{\boldsymbol{\xi}}_k + n_k^{j,L}$$

with  $n_k^{j,L} \sim (0, \sigma_L^2)$ .

## REFERENCES

- [1] L. Huang, "Review on LiDAR-based SLAM techniques," in *Proc. 2021 Int. Conf. Signal Process. Mach. Learn.*, 2021, pp. 163–168.
- [2] M. U. Khan, S. A. A. Zaidi, A. Ishaq, S. U. R. Bukhari, S. Samer, and A. Farman, "A comparative survey of LiDAR-SLAM and LiDAR based sensor technologies," in *Proc. 2021 Mohammad Ali Jinnah Univ. Int. Conf. Comput.*, 2021, pp. 1–8.
- [3] Q. Zou, Q. Sun, L. Chen, B. Nie, and Q. Li, "A comparative analysis of LiDAR SLAM-based indoor navigation for autonomous vehicles," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 7, pp. 6907–6921, Jul. 2022.
- [4] D. Cattaneo, M. Vaghi, and A. Valada, "LCDNet: Deep loop closure detection and point cloud registration for LiDAR SLAM," *IEEE Trans. Robot.*, vol. 38, no. 4, pp. 2074–2093, Aug. 2022.
- [5] C. Wang, Z. Cao, J. Li, J. Yu, and S. Wang, "Hierarchical distribution-based tightly-coupled LiDAR inertial odometry," *IEEE Trans. Intell. Veh.*, vol. 9, no. 1, pp. 1423–1435, Jan. 2024.
- [6] J. Chen, H. Wang, M. Hu, and P. N. Suganthan, "Versatile LiDAR-inertial odometry with SE (2) constraints for ground vehicles," *IEEE Robot. Automat. Lett.*, vol. 8, no. 6, pp. 3486–3493, Jun. 2023.
- [7] X. Xu et al., "A review of multi-sensor fusion SLAM systems based on 3D LiDAR," *Remote Sens.*, vol. 14, no. 12, 2022, Art. no. 2835.
- [8] 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, no. 7, Art. no. 2200459.
- [9] P. Shi, Z. Zhu, S. Sun, X. Zhao, and M. Tan, "Invariant extended Kalman filtering for tightly coupled LiDAR-inertial odometry and mapping," *IEEE/ASME Trans. Mechatron.*, vol. 28, no. 4, pp. 2213–2224, Aug. 2023.
- [10] 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. Automat. Lett.*, vol. 7, no. 2, pp. 4861–4868, Apr. 2022.
- [11] K. Huang, J. Zhao, Z. Zhu, C. Chen, and T. Feng, "LOG-LIO: A LiDAR-inertial odometry with efficient local geometric information estimation," *IEEE Robot. Automat. Lett.*, vol. 9, no. 1, pp. 459–466, Jan. 2024.
- [12] 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.
- [13] 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.
- [14] A. Barrau and B. Silvere, "The invariant extended Kalman filter as a stable observer," *IEEE Trans. Autom. Control*, vol. 62, no. 4, pp. 1797–1812, Apr. 2017.
- [15] C. Zhang, J. Qin, C. Yan, Y. Shi, Y. Wang, and M. Li, "Towards invariant extended Kalman filter-based resilient distributed state estimation for moving robots over mobile sensor networks under deception attacks," *Automatica*, vol. 159, 2024, Art. no. 111408.
- [16] J. Liu, Y. Zhang, X. Zhao, Z. He, W. Liu, and X. Lv, "Fast and robust LiDAR-inertial odometry by tightly-coupled iterated Kalman smoother and robocentric voxels," *IEEE Trans. Intell. Transp. Syst.*, vol. 25, no. 10, pp. 14486–14496, Oct. 2024, doi: [10.1109/TITS.2024.3391291](https://doi.org/10.1109/TITS.2024.3391291).
- [17] H. Zhang, R. Xiao, J. Li, C. Yan, and H. Tang, "A high-precision LiDAR-inertial odometry via invariant extended Kalman filtering and efficient surface mapping," *IEEE Trans. Instrum. Meas.*, vol. 73, 2024, Art. no. 8502911.
- [18] B. Axel and S. Bonnabel, "An EKF-SLAM algorithm with consistency properties," 2015, *arXiv:1510.06263*.
- [19] Z. Yan, L. Sun, T. Krajnik, and Y. Ruichek, "EU long-term dataset with multiple sensors for autonomous driving," in *Proc. 2020 IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2020, pp. 10697–10704.
- [20] J. Yin, A. Li, T. Li, W. Yu, and D. Zou, "M2DGR: A multi-sensor and multi-scenario SLAM dataset for ground robots," *IEEE Robot. Automat. Lett.*, vol. 7, no. 2, pp. 2266–2273, Apr. 2022.
- [21] 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. 2020 IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2020, pp. 5135–5142.