2023 IEEE International Conference on Robotics and Automation (ICRA 2023)
May 29 - June 2, 2023. London, UK

Continuous-Time LiDAR-Inertial-Vehicle Odometry Method with Lateral Acceleration Constraint

Bin He[1], Weichen Dai[2], Zeyu Wan[1], Hong Zhang[1] and Yu Zhang[1,3]

Abstract— **In this paper, we propose a continuous-time-based LiDAR-inertial-vehicle odometry method, which can tightly fuse the data from Light Detection And Ranging (LiDAR), inertial measurement units (IMU), and vehicle measurements. The lateral acceleration constraint is further added to trajectory estimation to make the estimated trajectory follow the motion characteristics of vehicles. In addition, since vehicle model parameters vary with different motion conditions and tyre pressure, we estimate vehicle correction factors that rectify changes in vehicle model parameters online, and also analyze the observability of these vehicle correction factors. In experiments, the proposed method is evaluated and compared with state-of-the-art methods in the public dataset. The experimental results show that the proposed method achieves more accurate results in all sequences since we add additional sensor measurements and utilize the characteristic of vehicle motion to restrict the trajectory estimation. The ablation study also proved the effectiveness of continuous-time representation, online correction factor estimation, and incorporation of lateral acceleration constraint.**

I. Introduction

In autonomous driving, localization is a prerequisite for decision making and high-definition map production. However, in GPS-denied environments such as tunnels and urban canyons, the high-precision GPS(RTK)/IMU navigation system (INS) is unreliable due to the multipath error. Thus, Simultaneous Location And Mapping (SLAM) methods that do not rely on external signals gain attention in autonomous driving, including methods based on visual cameras [1]–[3] and Light Detection And Ranging (LiDAR) sensors [4], [5]. Since visual solutions are very sensitive to lighting conditions and scene textures, LiDAR-based SLAM, which provides high-quality distance information, is widely used in practice.

To further reduce localization errors due to the limitation of a single type of sensor, besides LiDAR sensors, some studies incorporate multiple types of sensors, including inertial measurement units (IMU), steering angle sensors, and wheel speedometers [6]–[9]. The IMU can directly measure linear acceleration and angular velocity. Based on the vehicle model, the steering angle sensors and wheel speedometers can provide linear and angular velocity, known as vehicle measurements [10].

Due to the difference in the collection frequency of several sensors, the trajectory can be estimated in discrete-time and continuous-time representations. Most methods estimate the trajectory in the discrete-time representation to fuse the information from LiDAR, IMU, and vehicle measurements. However, taking advantage of the raw high-rate information of the vehicle measurements and IMU data is challenging, as the method based on the discrete-time representation requires many variables to be estimated if the raw data is utilized directly [11]. Thus, in the methods based on discrete-time representation, raw IMU and vehicle measurements have to be integrated to get relative pose measurements over a discrete-time period. Compared to the discrete-time representation, the continuous-time representation supports sampling poses at any timestamp. It has closed-form analytic derivatives, which allow direct registration of raw motion information obtained from high-rate vehicle measurements and IMU data. Therefore, the continuous-time representation has recently received much attention.

Moreover, the estimated trajectory of the vehicle should not only fit the measurements read from multiple onboard sensors mentioned above, but also follow the motion characteristics of the platform. The vehicle platform has a specific motion characteristic, lateral acceleration constraint, which shows the relationship between lateral acceleration, yaw rate, and forward speed. However, due to noise, the estimated trajectory, best fitted to the sensor measurements, may not be fitted with the vehicle motion characteristics. Thus, to make the estimated trajectory follow the motion characteristics of vehicles, such characteristics should be incorporated into the continuous-time trajectory estimation as constraints.

In this paper, following continuous-time representation, we present a continuous-time LiDAR-inertial-vehicle odometry method that tightly couples LiDAR, IMU, and vehicle measurements. The lateral acceleration constraint is incorporated into the continuous-time trajectory estimation through time sampling. Furthermore, since vehicle model parameters vary with different motion conditions and tyre pressure, we estimate correction factors for vehicle model parameters online and analyze the observability of these correction factors in the system. In experiments, the proposed method is compared to state-of-the-art methods to evaluate the accuracy. Furthermore, an ablation study is used to show the effectiveness of continuous-time-based LiDAR-inertial-vehicle fusion, online estimation of correction factors, and

This work was supported by STI 2030-Major Projects 2021ZD0201403, in part by NSFC 62088101 Autonomous Intelligent Unmanned Systems, and in part by the Open Research Project of the State Key Laboratory of Industrial Control Technology, Zhejiang University, China (No.ICT2022B04), Zhejiang Provincial Natural Science Foundation of China under Grant No.LQ22F030022.*(Corresponding author: Yu Zhang.)*

[1]State Key Laboratory of Industrial Control Technology, College of Control Science and Engineering, Zhejiang University, Hangzhou 310000, China

[2]School of Computer Science, Hangzhou Dianzi University, Hangzhou 310018, China

[3]Key Laboratory of Collaborative sensing and autonomous unmanned systems of Zhejiang Province, Hangzhou 310007, China

E-mail: {binhe, and zhangyu80}@zju.edu.cn

the incorporation of the lateral acceleration constraint.

We summarize the contributions of this paper as follows:

- We propose a continuous-time-based LiDAR-inertial-vehicle odometry method, which can tightly fuse the LiDAR scans, IMU data, vehicle measurements, and vehicle motion characteristics constraint.
- We propose introducing the lateral acceleration constraint into trajectory estimation to provide a trajectory constraint.
- We estimate the correction factors of the vehicle model parameters online. Furthermore, their observability is further analyzed in this paper.

II. Related Work

In recent years, studies on LiDAR-based multi-sensor odometry have made significant progress. In IMU-aided LOAM [4] and LeGO-LOAM [5], IMU is mainly taken to give an initial motion estimate for LiDAR scan matching. Zuo et al. [12] fuse the LiDAR scan registration with IMU and visual measurements using the Multi-State Constraint Kalman Filter (MSCKF). Zhen et al. [13] use error-state Kalman Filter (ESKF) to combine the registration results and IMU measurements. The methods above can be classified as loosely-coupled methods. Loosely-coupled methods typically process scan registration and inertial measurements separately. Therefore, they are computationally efficient and flexible for sensor incorporation, but less accurate. In contrast to the loosely coupled approaches, several tightly-coupled methods fuse the LiDAR measurements directly, instead of only utilizing point cloud registration results, with other sensor measurements. LIO-mapping [14] utilizes a graph optimization based on IMU preintegration constraints and LiDAR edge/plane features. LIO-SAM [15] incorporates LiDAR measurements, IMU preintegration, and optional GNSS measurements into the system using a factor graph. FAST-LIO2 [16] directly register the raw LiDAR scan to map without extracting feature points, and fuse all measurements based on the Iterated Kalman Filter.

Recent works have also considered additional information from sensor platforms, such as vehicle platform information, to enhance LIO performance. Júnior et al. [7] fuse wheel odometry and IMU data into LiDAR odometry using EKF. In [9], an odometer increment model is introduced to fuse IMU and encoder measurements. LIO-Vehicle [6] combines LiDAR-inertial odometry with vehicle dynamics measurements preintegration. Hess et al. [8] build a pose graph to couple LiDAR scans, IMU integration, odometer observation, and GNSS data. However, these methods do not incorporate constraints on vehicle motion characteristics into trajectory estimation. Yu et al. [17] and Huang et al. [18] take vehicle velocity direction information, but do not consider the phenomenon of vehicle side slip and cannot provide additional information when vehicle measurements are available.

In addition to the methods in a discrete-time fashion, some studies also employ continuous-time representations

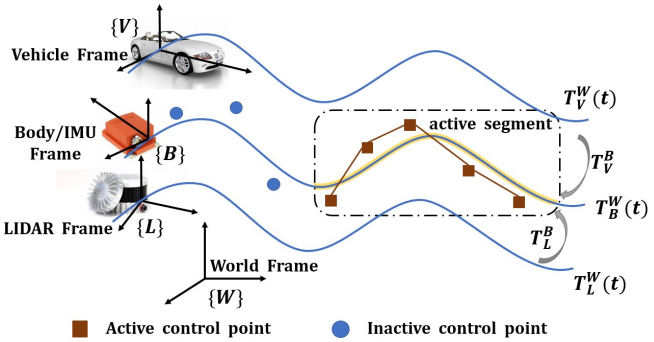

Fig. 1: The definition of the involved frames and the illustration of definitions involved in cumulative B-spline-based continuous-time trajectory. The trajectory of LiDAR $\mathbf{T}_L^W(u(t))$ and vehicle $\mathbf{T}_V^W(u(t))$ can be calculated by the trajectory of IMU $\mathbf{T}_B^W(u(t))$ and the corresponding extrinsic transformation $\mathbf{T}_L^B$ and $\mathbf{T}_V^B$.

to address the LiDAR distortion problem or fuse measurements obtained from LiDAR and IMU sensors at arbitrary frequencies. Continuous-ICP [19] uses B-spline basis functions to represent the trajectory and takes the movement during LiDAR scan collection into account. Droeschel et al. [20] present a continuous-time LiDAR-only SLAM method based on a hierarchical refinement back-end. Cong et al. [21] integrate a spline motion model with non-rigid LiDAR point cloud fusion for continuous mapping. CLINS [22] is a continuous-time-based LiDAR-inertial system with a two-stage trajectory correction. Park et al. [23] utilize a continuous-time map deformation method to maintain a globally consistent map and eliminates the need for global trajectory optimization.

III. The Proposed Methodology

We define the frames and notation used throughout the paper. We use the notation $(\cdot)^W$ to represent the world reference frame W, and $(\cdot)^B$, $(\cdot)^L$, $(\cdot)^V$ are the quantities in the IMU body frame, the LiDAR frame, and the vehicle frame (see Fig.1). $\mathbf{R}_B^A \in SO(3)$ and $\mathbf{p}_B^A \in \mathbb{R}^3$ is the orientation and position of the frame $\{B\}$ with respect to $\{A\}$.

A. Representation of Continuous-Time Trajectory

B-splines have the smoothness and local support property (a value of order k spline is controlled only by a local subset of the control points). As in [24], [25], we use two separate B-splines to represent the translation in $\mathbb{R}^3$ and the rotation in the Lie group $SO(3)$, respectively. The sampling position of the order k cumulative B-spline at time t can be written as

$$\mathbf{p}\big(u(t)\big) = \mathbf{p}_i + \sum_{j=1}^{k-1} \lambda_j\big(u(t)\big) \cdot \mathbf{d}_j^i \tag{1}$$

As derivative of the B-spline encoding the trajectory positions, the velocity and acceleration in the global frame are defined by

$$\mathbf{v}\big(u(t)\big) = \sum_{j=1}^{k-1} \dot{\lambda}_j\big(u(t)\big) \cdot \mathbf{d}_j^i \tag{2}$$

$$\mathbf{a}\big(u(t)\big) = \sum_{j=1}^{k-1} \ddot{\lambda}_j\big(u(t)\big) \cdot \mathbf{d}_j^i \tag{3}$$

with $u(t) = \frac{t-t_0}{\Delta t} - i$ and $t \in [t_i, t_{i+1})$. The coefficients $\lambda_j\big(u(t)\big) \in \mathbb{R}$ depend on the order of the B-spline [26]. $\mathbf{d}_j^i = \mathbf{p}_{i+j} - \mathbf{p}_{i+j-1}$ is the distance vector.

For $\mathbf{R} \in SO(3)$, the formulation of cumulative B-spline is defined as

$$\mathbf{R}\big(u(t)\big) = \mathbf{R}_i \cdot \prod_{j=1}^{k-1} \mathrm{Exp}(\lambda_j\big(u(t)\big) \cdot \mathbf{d}_j^i) \tag{4}$$

$$\boldsymbol{\omega}\big(u(t)\big) = [\mathbf{R}^T \dot{\mathbf{R}}]_\vee \tag{5}$$

$$\dot{\boldsymbol{\omega}}\big(u(t)\big) = [\mathbf{R}^T \ddot{\mathbf{R}} - \boldsymbol{\omega}_\wedge^2]_\vee \tag{6}$$

where $\mathbf{d}_j^i = \mathrm{Log}(\mathbf{R}_{\mathrm{i+j-1}}^{-1} \cdot \mathbf{R}_{\mathrm{i+j}})$. $\mathrm{Exp}(\cdot)$ is the exponential matrix that maps the tangent vectors to the Lie group, and the inverse operation is denoted by $\mathrm{Log}(\cdot)$. $[\cdot]_\wedge$ maps a vector to the skew-symmetric matrix, and the inverse operation is $[\cdot]_\vee$.

The value of the continuous-time trajectory in the global frame $\mathbf{T}\big(u(t)\big) = \{\mathbf{R}\big(u(t)\big),\ \mathbf{p}\big(u(t)\big)\}$, with $t \in [t_i, t_{i+1})$, only depends on the active control points set $\boldsymbol{\Phi}(t_i, t_{i+1}) = \{\boldsymbol{\phi}_i, \ldots, \boldsymbol{\phi}_{i+k-1}\}$ (shown in Fig.1). The j-th control point in the set of control points is $\boldsymbol{\phi}_{i+j} := \{\mathbf{R}_{i+j},\ \mathbf{p}_{i+j}\}$.

B. Bicycle Model of the Vehicle

Since vehicles are affected by the tyre sideslip phenomenon, the direction of vehicle velocity is not always along the longitudinal direction, especially when the vehicle turns. Therefore, we utilize the bicycle model [27] which takes vehicle sideslip phenomenon into account (shown in Fig.2). Considering the steady-state steering motion of the vehicle, the basic equations for the bicycle model are as follows.

$$C_f(\delta - \alpha - \frac{l_f\omega}{v_x}) + C_r(\frac{l_r\omega}{v_x} - \alpha) = mv_x\omega \tag{7}$$

$$l_f C_f(\delta - \alpha - \frac{l_f\omega}{v_x}) - l_r C_r(\frac{l_r\omega}{v_x} - \alpha) = 0 \tag{8}$$

where m is the mass of the vehicle, C_f and C_r are the lateral stiffness of the front and rear wheels.

From (7) and (8), the vehicle deflection angle α can be obtained by using

$$\alpha = \frac{\frac{1}{\tilde{v}^2} l l_r C_f C_r + \frac{1}{2} m C_f l_f}{l^2 C_f C_r - m\tilde{v}^2(l_f C_f - l_r C_r)} \tilde{v}^2\delta = D\tilde{v}^2\delta \tag{9}$$

where D is the model deflection parameter, which is corrected online (as detailed in Sec.III-C). The yaw rate is

$$\tilde{\omega} = \frac{\tilde{v} l C_r C_f \delta}{l^2 C_r C_f + m l_f C_f \tilde{v}^2 - m l_r C_r \tilde{v}^2} \tag{10}$$

where $l = l_f + l_r$, δ is the angle of the steering wheel obtained from the steering angle sensors, and $\tilde{v}$ is the reading of the wheel speedometer.

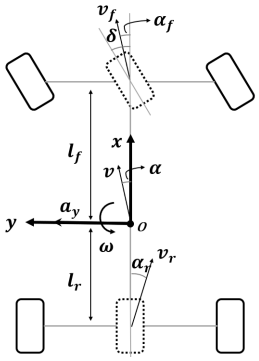

Fig. 2: The bicycle model considering the sideslip phenomenon. The center of mass O is defined as the origin of the vehicle frame. The x and y axis are defined along the vehicle's longitudinal and lateral directions.

C. Vehicle Measurements and observability of Correction Factors

As vehicles usually travel in various situations, including those with different motion conditions and tyre pressure, the parameters of the vehicle bicycle model would change. The raw vehicle measurements will be affected. Thus, the changed vehicle model parameters should be corrected. Moreover, the observability should be analyzed since the correction factors may not be correctly estimated under certain degenerate motion conditions.

We use the factor $c = [c_v, c_\alpha, c_\omega]^T$ to correct the vehicle bicycle model, including the linear velocity magnitude, model deflection parameters, and angular velocity. From (9) and (10), the vehicle measurements are given by

$$\begin{aligned} \hat{\mathbf{v}}^V &= [c_v\tilde{v}\cos(c_\alpha\tilde{\alpha}) \quad \tilde{v}\sin(c_\alpha\tilde{\alpha}) \quad 0]^T \\ \hat{\boldsymbol{\omega}}^V &= [0 \quad 0 \quad c_\omega\tilde{\omega}]^T \end{aligned} \tag{11}$$

where $\tilde{\alpha} = \tilde{v}^2\delta$. Note that the initial guess of the correction factor is set to $[1, D, 1]^T$, with D being the coefficient in (9).

We further analyze the observability of correction factors in this system. It demonstrates when the factors can be correctly estimated. We follow the methodology in [28] and [29]. Supposing that we have pre-measured external parameters $\mathbf{T}_V^B = \{\mathbf{p}_V^B, \mathbf{R}_V^B\}$ between the IMU body frame and the vehicle dynamic frame, we can get the transformation from the body frame to the vehicle frame as follows:

$$\mathbf{v}_V^W = \mathbf{v}_B^W + \mathbf{R}_B^W [\boldsymbol{\omega}^B]_\wedge \mathbf{p}_V^B \tag{12}$$

$$\mathbf{a}_V^W = \mathbf{a}_B^W + \mathbf{R}_B^W [\dot{\boldsymbol{\omega}}^B]_\wedge \mathbf{p}_V^B + \mathbf{R}_B^W [\boldsymbol{\omega}^B]_\wedge^2 \mathbf{p}_V^B \tag{13}$$

From (12)-(13), we can get the constraint between the vehicle measurements $\hat{\mathbf{v}}^V, \hat{\boldsymbol{\omega}}^V$ and the inferred measurements $\mathbf{R}_V^{B^T}\check{\mathbf{v}}^B, \mathbf{R}_V^{B^T}\check{\boldsymbol{\omega}}^B$ obtained from the estimated trajectory:

$$\begin{aligned} \hat{\mathbf{v}}^V &= -[\mathbf{R}_V^{B^T}\check{\boldsymbol{\omega}}^B]_\wedge \mathbf{p}_B^V + \mathbf{R}_V^{B^T}\check{\mathbf{v}}^B \\ \hat{\boldsymbol{\omega}}^V &= \mathbf{R}_V^{B^T} \cdot \check{\boldsymbol{\omega}}^B \end{aligned} \tag{14}$$

Substituting (11) into (14), the constraint equation matrix at time t is formed as

$$\mathcal{M}(\mathrm{t}) = \begin{bmatrix} [\mathbf{R}_V^{B^T}\check{\mathbf{v}}^B]_x + [\mathbf{R}_V^{B^T}\check{\boldsymbol{\omega}}^B]_z [\mathbf{p}_B^V]_y - c_v\tilde{v}cos(c_\alpha\tilde{\alpha}) \\ [\mathbf{R}_V^{B^T}\check{\mathbf{v}}^B]_y - [\mathbf{R}_V^{B^T}\check{\boldsymbol{\omega}}^B]_z [\mathbf{p}_B^V]_x - \tilde{v}sin(c_\alpha\tilde{\alpha}) \\ [\mathbf{R}_V^{B^T}\check{\boldsymbol{\omega}}^B]_z - c_\omega\tilde{\omega} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \tag{15}$$

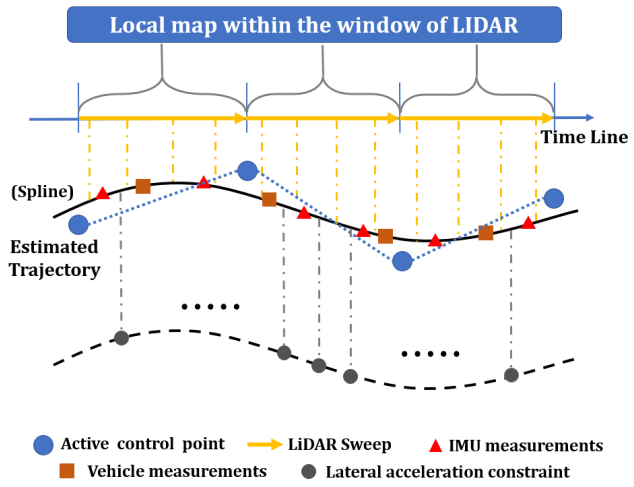

Fig. 3: Concept of our algorithm. The LiDAR scans, IMU data, vehicle measurements, and lateral acceleration constraint are incorporated to optimize the active control points and thus obtain the estimated trajectory.

Within a set time period $\mathcal{S} = \{\mathrm{t}_0, \mathrm{t}_1, \cdots, \mathrm{t}_\mathrm{s}\}$, the observability matrix can be formed as:

$$\mathcal{O} = \left[\frac{\partial \mathcal{M}(\mathrm{t}_0)}{\partial c} \quad \frac{\partial \mathcal{M}(\mathrm{t}_1)}{\partial c} \quad \cdots \quad \frac{\partial \mathcal{M}(\mathrm{t}_\mathrm{i})}{\partial c}\right]^T = \begin{bmatrix} -\tilde{v}(t_0)cos(c_\alpha\tilde{\alpha}(t_0)) & \tilde{\alpha}(t_0)c_v\tilde{v}(t_0)sin(c_\alpha\tilde{\alpha}(t_0)) & 0 \\ 0 & -\tilde{\alpha}(t_0)c_v\tilde{v}(t_0)cos(c_\alpha\tilde{\alpha}(t_0)) & 0 \\ 0 & 0 & -\tilde{\omega}(t_0) \\ \vdots & \vdots & \vdots \\ -\tilde{v}(t_i)cos(c_\alpha\tilde{\alpha}(t_i)) & \tilde{\alpha}(t_i)c_v\tilde{v}(t_i)sin(c_\alpha\tilde{\alpha}(t_i)) & 0 \\ 0 & -\tilde{\alpha}(t_i)c_v\tilde{v}(t_i)cos(c_\alpha\tilde{\alpha}(t_i)) & 0 \\ 0 & 0 & -\tilde{\omega}(t_i) \end{bmatrix} \tag{16}$$

The correction factors c are locally observable when the observability matrix has the full column rank.

To investigate the degenerate cases of the correction factor estimation, we inspect whether there is a non-zero vector k such that $\mathcal{O}\mathbf{k} = \mathbf{0}, \mathbf{k} = [\mathrm{k}_1, \mathrm{k}_2, \mathrm{k}_3]^\mathrm{T} \neq \mathbf{0}$. Such a k exists when one of the following conditions is satisfied:

- $\tilde{\omega}(t) = 0$, $\mathbf{k} = [0, 0, \rho]^\mathrm{T}$
- $\tilde{v}(t) = 0$, $\mathbf{k} = [\rho_1, \rho_2, 0]^\mathrm{T}$

where ρ is any non-zero value. Therefore, the correction factors can be correctly estimated except for the following degenerate cases: c_α, c_v are unobservable when the wheel speed is nearly zero (the vehicle stops), c_ω is unobservable when the angular velocity obtained from vehicle measurements is nearly zero (the vehicle moves straight forward).

D. Estimator Formulation

We denote the states to be estimated during a LiDAR scan S_k and the local window as:

$$\boldsymbol{\chi}_k = \left(\boldsymbol{\Phi}(t_k, t_k + \Delta T),\ \mathbf{b}_{ak},\ \mathbf{b}_{g_k},\ c_{vk},\ c_{\alpha k},\ c_{\omega k}\right) \tag{17}$$

$$\hat{\boldsymbol{\chi}} = \left(\boldsymbol{\chi}_k,\ \boldsymbol{\chi}_{k+1}\ \cdots\ \boldsymbol{\chi}_{k+M}\right) \tag{18}$$

where t_k is the timestamp at the beginning of the scan S_k, ΔT is the time it takes to complete a LiDAR scan; $\boldsymbol{\Phi}(t_k, t_k + \Delta T)$ is the control points set during the scan S_k; $\mathbf{b}_{ak}, \mathbf{b}_{g_k} \in \mathbb{R}^3$ are the bias of accelerator and gyroscope; $c_{vk}, c_{\alpha k}, c_{\omega k} \in \mathbb{R}$ are the correction factors. M is the number of LiDAR scans in the local window. Here, M is set to 5.

Fig. 3 illustrates the concept of our algorithm, which combines IMU, vehicle measurements, LiDAR sweep, and lateral acceleration constraint to estimate the trajectory. The cost function can be constructed as

$$\arg\min_{\hat{\boldsymbol{\chi}}} \left\{ \sum \|\mathbf{r}_\mathcal{L}\|^2_{\Sigma_\mathcal{L}} + \sum \|\mathbf{r}_\mathcal{I}\|^2_{\Sigma_\mathcal{I}} + \sum \|\mathbf{r}_\mathcal{V}\|^2_{\Sigma_\mathcal{V}} + \sum \|\mathbf{r}_\mathcal{M}\|^2 \right\} \tag{19}$$

where $\mathbf{r}_\mathcal{L}$, $\mathbf{r}_\mathcal{I}$, $\mathbf{r}_\mathcal{V}$ are residuals constructed from LiDAR, IMU, vehicle measurements. $\mathbf{r}_\mathcal{M}$ is the lateral acceleration constraint residual. $\|\cdot\|^2$ is the Mahalanobis norm with weighted covariance $\boldsymbol{\Sigma}$. The LiDAR term $\mathbf{r}_\mathcal{L}$ used in this paper is similar to the existing work [21], [22]. We adopt the LM method of Ceres [30] to solve this problem.

E. Vehicle Measurements and IMU Components

B-spline has closed-form analytic derivatives. Therefore, the B-spline can match with each raw high-rate vehicle measurements and IMU data without requiring integration. The prediction at time t_V, comprising vehicle linear and angular velocity, can be computed from the estimated body trajectory:

$$\begin{aligned} \mathbf{v}^V\left(u(t_V)\right) =& \mathbf{R}_V^{B^T}\mathbf{R}_B^W\left(u(t_V)\right)^T\mathbf{v}_B^W\left(u(t_V)\right) \\ &+ \mathbf{R}_V^{B^T}\left[\boldsymbol{\omega}^B\left(u(t_V)\right)\right]_\wedge \mathbf{p}_V^B \end{aligned} \tag{20}$$

$$\boldsymbol{\omega}^V\left(u(t_V)\right) = \mathbf{R}_V^{B^T}\boldsymbol{\omega}^B\left(u(t_V)\right) \tag{21}$$

Considering the vehicle measurements within two consecutive LiDAR scans S_γ and $S_{\gamma+1}$ in the local window, the vehicle measurement residual is defined as

$$\mathbf{r}_\mathcal{V} = \begin{bmatrix} \mathbf{v}^V\left(u(t_V)\right) - \hat{\mathbf{v}}^V \\ \boldsymbol{\omega}^V\left(u(t_V)\right) - \hat{\boldsymbol{\omega}}^V \\ c_{v\gamma+1} - c_{v\gamma} \\ c_{\alpha\gamma+1} - c_{\alpha\gamma} \\ c_{\omega\gamma+1} - c_{\omega\gamma} \end{bmatrix} \tag{22}$$

where t_V is the timestamp of the vehicle measurement; $\hat{\mathbf{v}}^V$, $\hat{\boldsymbol{\omega}}^V$ are the corresponding vehicle measurement as described in (11).

The residual constructed from the accelerometer and gyroscope measurements can be formed as

$$\mathbf{r}_\mathcal{I} = \begin{bmatrix} \mathbf{R}_B^W\left(u(t_I)\right)^T\left(\mathbf{a}_B^W\left(u(t_I)\right) + \mathbf{g}^W\right) - \hat{\mathbf{a}}^I + \mathbf{b}_{a\gamma} \\ \boldsymbol{\omega}^B\left(u(t_I)\right) - \hat{\boldsymbol{\omega}}^I + \mathbf{b}_{g_\gamma} \\ \mathbf{b}_{a\gamma+1} - \mathbf{b}_{a\gamma} \\ \mathbf{b}_{g_{\gamma+1}} - \mathbf{b}_{g_\gamma} \end{bmatrix} \tag{23}$$

where t_I is the timestamp of the raw IMU measurements $\hat{\mathbf{a}}^I$ and $\hat{\boldsymbol{\omega}}^I$; The vector $\mathbf{g}^W \in \mathbb{R}^3$ is gravity.

F. Lateral Acceleration Constraint

In addition to fitting the sensor readings, the estimated trajectory of the vehicle should follow the motion characteristics of the vehicle platform. However, due to the sensor noise, the estimated trajectory that is best fitted to the sensor measurements may not be consistent with vehicle motion characteristics. To make the estimated trajectory follow the motion characteristics of the vehicle, the vehicle motion characteristics should be incorporated into the trajectory estimation as a trajectory constraint.

Therefore, we propose the lateral acceleration constraint, which is motivated by the lateral acceleration equation in the vehicle frame. As shown in [27], since the vehicle can be treated as making a steady-state motion, the lateral acceleration equation can be described as:

$$[\mathbf{a}^V]_y = [\mathbf{v}^V]_x [\boldsymbol{\omega}^V]_z \tag{24}$$

where $[\cdot]_x$, $[\cdot]_y$, $[\cdot]_z$ represent the components along the xyz-axis and the corresponding axial angle rates.

In order to express (24) in terms of the estimated trajectory variables, the linear velocity, linear acceleration, and angular velocity of the vehicle trajectory at sampling time t_s can be obtained through:

$$\begin{aligned}\mathbf{v}^V\big(u(t_s)\big) =& \mathbf{R}_V^{B^T}\mathbf{R}_B^W\big(u(t_s)\big)^T\mathbf{v}_B^W\big(u(t_s)\big) \\ &+ \mathbf{R}_V^{B^T}\big[\hat{\boldsymbol{\omega}}^I - \mathbf{b}_g\big]_\wedge \mathbf{p}_V^B\end{aligned} \tag{25}$$

$$\begin{aligned}\mathbf{a}^V\big(u(t_s)\big) =& \ \mathbf{R}_V^{B^T}\mathbf{R}_B^W\big(u(t_s)\big)^T\mathbf{a}_B^W\big(u(t_s)\big) \\ &+ \mathbf{R}_V^{B^T}[\hat{\boldsymbol{\omega}}^I - \mathbf{b}_g]_\wedge^2\mathbf{p}_V^B + \mathbf{R}_V^{B^T}[\dot{\boldsymbol{\omega}}^B\big(u(t_s)\big)]_\wedge\mathbf{p}_V^B\end{aligned} \tag{26}$$

$$\hat{\boldsymbol{\omega}}_m^V = \mathbf{R}_V^{B^T}\cdot(\hat{\boldsymbol{\omega}}^I - \mathbf{b}_g) \tag{27}$$

where $\hat{\boldsymbol{\omega}}^I$ is the measurements of the gyroscope. $\hat{\boldsymbol{\omega}}_m^V$ is the inferred measurements in the vehicle frame obtained from the gyroscope measurements. In practice, we set the sampling time t_s as the IMU timestamp t_I.

The lateral acceleration constraint residual at the sampling time t_s can be formed as:

$$\mathbf{r}_\mathcal{M} = \mathbf{U}\Big(\mathbf{a}^V\big(u(t_s)\big) - \big[[\hat{\boldsymbol{\omega}}_m^V]_z\big]_\wedge \mathbf{v}^V\big(u(t_s)\big)\Big) \tag{28}$$

with the selective matrix $\mathbf{U} = [0,\ 1,\ 0]$. The Jacobian of the lateral acceleration constraint residual $\mathbf{r}_\mathcal{M}$ with respect to the control points $\mathbf{R}_{i+j}$ and $\mathbf{p}_{i+j}$ should be derived since it is necessary for batch optimization. Thus, we have the following derivation through the chain rule:

$$\begin{aligned}\frac{\partial \mathbf{r}_\mathcal{M}}{\partial \mathbf{R}_{i+j}} =& \frac{\partial \mathbf{r}_\mathcal{M}}{\partial \mathbf{R}_B^W}\cdot\Big(\frac{\partial \mathbf{R}_B^W}{\partial \mathbf{d}_j^i}\cdot\frac{\partial \mathbf{d}_j^i}{\partial \mathbf{R}_{i+j}} + \frac{\partial \mathbf{R}_B^W}{\partial \mathbf{d}_{j+1}^i}\cdot\frac{\partial \mathbf{d}_{j+1}^i}{\partial \mathbf{R}_{i+j}}\Big) \\ &+ \frac{\partial \mathbf{r}_\mathcal{M}}{\partial \dot{\boldsymbol{\omega}}^B}\cdot\Big(\frac{\partial \dot{\boldsymbol{\omega}}^B}{\partial \mathbf{d}_j^i}\cdot\frac{\partial \mathbf{d}_j^i}{\partial \mathbf{R}_{i+j}} + \frac{\partial \dot{\boldsymbol{\omega}}^B}{\partial \mathbf{d}_{j+1}^i}\cdot\frac{\partial \mathbf{d}_{j+1}^i}{\partial \mathbf{R}_{i+j}}\Big)\end{aligned} \tag{29}$$

$$\begin{aligned}\frac{\partial \mathbf{r}_\mathcal{M}}{\partial \mathbf{p}_{i+j}} =& \frac{\partial \mathbf{r}_\mathcal{M}}{\partial \mathbf{a}_B^W}\cdot\Big(\frac{\partial \mathbf{a}_B^W}{\partial \mathbf{d}_j^i}\cdot\frac{\partial \mathbf{d}_j^i}{\partial \mathbf{p}_{i+j}} + \frac{\partial \mathbf{a}_B^W}{\partial \mathbf{d}_{j+1}^i}\cdot\frac{\partial \mathbf{d}_{j+1}^i}{\partial \mathbf{p}_{i+j}}\Big) \\ &+ \frac{\partial \mathbf{r}_\mathcal{M}}{\partial \mathbf{v}_B^W}\cdot\Big(\frac{\partial \mathbf{v}_B^W}{\partial \mathbf{d}_j^i}\cdot\frac{\partial \mathbf{d}_j^i}{\partial \mathbf{p}_{i+j}} + \frac{\partial \mathbf{v}_B^W}{\partial \mathbf{d}_{j+1}^i}\cdot\frac{\partial \mathbf{d}_{j+1}^i}{\partial \mathbf{p}_{i+j}}\Big)\end{aligned} \tag{30}$$

with $\mathbf{R}_{i+j}$ and $\mathbf{p}_{i+j}$, as detailed in Sec. III-A, are the control points of the rotation trajectory and the translation trajectory, respectively.

The elements in (29) and (30) are computed as:

$$\frac{\partial \mathbf{r}_\mathcal{M}}{\partial \mathbf{R}_B^W} = \mathbf{U}\Big(\mathbf{R}_V^{B^T}\mathbf{R}_B^{W^T}[\mathbf{a}_B^W]_\wedge - \big[[\hat{\boldsymbol{\omega}}_m^V]_z\big]_\wedge\mathbf{R}_V^{B^T}\mathbf{R}_B^{W^T}[\mathbf{v}_B^W]_\wedge\Big) \tag{31}$$

$$\frac{\partial \mathbf{r}_\mathcal{M}}{\partial \dot{\boldsymbol{\omega}}^B} = -\mathbf{U}\mathbf{R}_V^{B^T}[\mathbf{p}_V^B]_\wedge \tag{32}$$

$$\frac{\partial \mathbf{r}_\mathcal{M}}{\partial \mathbf{a}_B^W} = \mathbf{U}\mathbf{R}_V^{B^T}\mathbf{R}_B^{W^T} \tag{33}$$

$$\frac{\partial \mathbf{r}_\mathcal{M}}{\partial \mathbf{v}_B^W} = -\mathbf{U}\big[[\hat{\boldsymbol{\omega}}_m^V]_z\big]_\wedge\mathbf{R}_V^{B^T}\mathbf{R}_B^{W^T} \tag{34}$$

the terms $\frac{\partial \mathbf{R}_B^W}{\partial \mathbf{d}_j^i}$ and $\frac{\partial \dot{\boldsymbol{\omega}}^B}{\partial \mathbf{d}_j^i}$ were derived in [26] via a recursive scheme.

IV. Experiments

In this section, the performance of our method is evaluated. We compare our method with other algorithms and perform an ablation study to test the impact of the continuous-time representation, online correction factor estimation, and incorporation of lateral acceleration constraint. We also show the estimation results of the correction factors, which validate the observability analysis. The algorithm is tested on a computer equipped with an AMD r5-1600 CPU.

A. Dataset

We evaluate our method on the publicly available dataset provided in [6], since it is the only public dataset that contains measurements of both wheel speed and steering angle. The LIO-Vehicle dataset [6] is collected from an autonomous vehicle operating on the Jiading campus of Tongji University. It captures LiDAR data (Velodyne VLP-16) and IMU measurements (Xsens MTi-30 AHRS, 9 Axis). The readings of the wheel speed and steering angle sensors are obtained at 100 Hz. The dataset contains several sequences in diverse scenarios, including big, small and LiDAR-degraded scenarios. It collected data at different speeds, which allows us to evaluate the performance under different motion conditions [6]. Table I lists the distance and average speed of the sequences Big and $Small$. Note that the vehicle travels along a small circle three times in the sequence $Small3$.

B. Comparison with Existing Methods

We compare our algorithm with existing algorithms. We use LIO-SAM [15] and CLINS [22] to represent the LiDAR-inertial odometry based on discrete and continuous time, and LIO-Vehicle [6] to represent the LiDAR-inertial odometry based on discrete time aided by vehicle measurements. Note that the experimental results for LIO-Vehicle [6] are acquired directly from the paper since it is not open source. The corresponding parameters in all the methods are set to the same. For the outdoor dataset, we remove the loop closure detection and only compare the front-end odometry of different methods. The resulting translational errors of each method are computed. Table I summarizes the root mean square error (RMSE) of the estimation from different

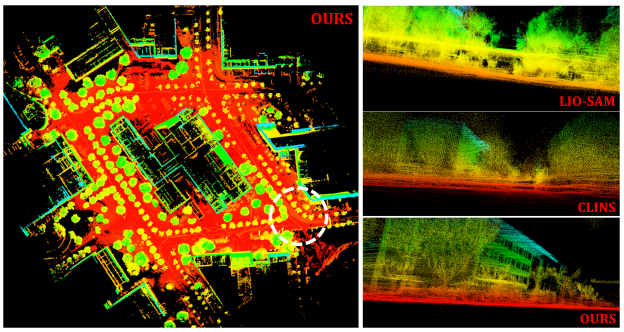

Fig. 4: Mapping result comparison using the Small3 dataset.

TABLE I: RMSE OF TRANSLATIONAL ERRORS(m) ON OUTDOOR DATASET

Datasets	Length(m)	Speed (km/h)	LIO-SAM	CLINS	LIO-Vehicle	Ours only Vehicle	Ours with CF	Ours CF+MC
Big 1	855.21	11.08	1.348698	1.144285	1.072767	0.864758	0.822656	**0.637822**
Big 2	828.98	22.65	1.694747	1.089169	1.196296	0.771056	0.764774	**0.550898**
Small 1	557.89	11.53	1.10915	0.850058	0.797489	0.738651	0.706639	**0.516458**
Small 2	544.235	28.72	1.455033	1.32291	0.783883	0.539993	0.484828	**0.380288**
Small 3	1643.36	11.47	2.038323	2.54067	1.533271	0.472946	0.483840	**0.435945**

TABLE II: ERRORS W.R.T GROUNDTRUTH ON TUNNEL DATASET

Method	LIO-SAM	LIO-SAM_lc	CLINS	CLINS_lc	LIO-Vehicle	LIO-Vehicle_lc	Ours	Ours_lc
RMSE (m)	16.042	7.28	15.702	8.082	4.978	3.726	2.949	1.798

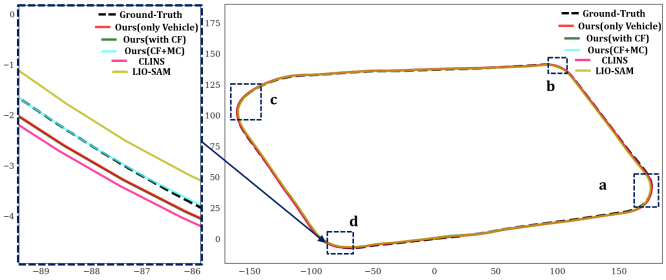

Fig. 5: Trajectory comparison using Big2 sequence.

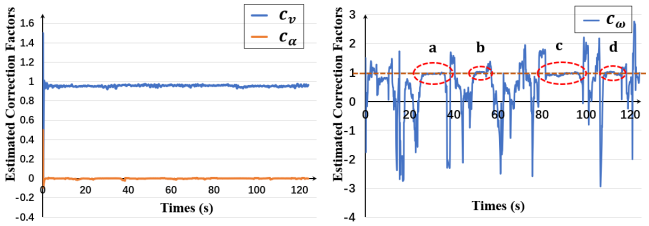

Fig. 6: Estimation of the correction factors on Big2 sequence.

methods. We also compare the mapping results between our method and existing methods using the $Small3$ sequence. As shown in Fig.4, the mapping result of our method is still clear even if the vehicle passes through the same place multiple times, while the mapping results of LIO-SAM and CLINS suffer from ghosting heavily. For the tunnel dataset, different methods are compared without/with loop closure (lc). The relative trajectory error (RTE) between the end of the estimated trajectory and GNSS localization is computed for evaluation (as shown in Table II). From Table I and Table II, we can see that the proposed method achieves more accurate estimation in all sequences, since we add additional sensor measurements into the trajectory estimation and provide trajectory constraints by utilizing the vehicle motion characteristic.

C. Ablation Study

We also demonstrate the effect of continuous-time representation, online estimation of correction factors, and incorporation of the proposed lateral acceleration constraint in our method. **OURS only Vehicle** is the limited version that only incorporates the vehicle measurements with IMU data and LiDAR features in a continuous-time fashion for the trajectory estimation. **OURS with CF** estimated the correction factors based on the above. **OURS CF+MC** is our complete algorithm with vehicle measurements, online estimation of correction factors, and lateral acceleration constraint incorporation. All other parameters of these methods are kept the same. The results are shown in Table I. Comparing **OURS only Vehicle** with LIO-vehicle, we find that the continuous-time representation performs better than discrete-time when fusing LiDAR features with multiple high-rate motion measurements (such as IMU and vehicle measurements). The results also show that the online estimation of correction factors and the incorporation of the lateral acceleration constraint can improve performance. The effect of adding vehicle lateral acceleration constraints is more pronounced. The trajectories of various methods using the sequence $Big2$ are shown in Fig. 5.

D. Estimation of Correction factors

In this section, we evaluate the convergence properties of correction factors. We add additional error terms [0.5, 0.5, 0.5] to the reasonable initial guesses of correction factors. As shown in Fig.6, the correction factors of deflection parameter and wheel speed (c_α, c_v) are quickly converged since the vehicle is always moving in this sequence. Regarding the angular velocity correction factor c_ω, it converges only in four time periods (as shown in the red circle in Fig.6). From Fig.5, there are four turning angles in the sequence $Big2$. The angular velocity measurements are noticeable when the vehicle is turning at these four corners, and the rest vehicle motion is mainly straight ahead. Thus, c_ω only converges when the vehicle passes through these four corners (corresponding a, b, c, d shown in Fig.5 and Fig.6). Vehicles take less time to pass through obtuse angles and more time to pass through acute angles in the sequence. The duration of the vehicle passing the corner is almost the same as the convergence time of c_ω. The results meet the theoretical expectations described in Sec. III-C. So we recommend not to estimate c_ω when the angular velocity is very small.

V. CONCLUSIONS

In this paper, we proposed a tightly coupled continuous-time LiDAR-inertial-vehicle trajectory estimator. We incorporate the vehicle motion characteristics into the trajectory estimation to provide trajectory constraints. The correction factors of the model parameters are also estimated. Moreover, their observability is analyzed to guide the estimation. The experimental results validate our observability analysis and also indicate that the inclusion of the lateral acceleration constraint can significantly improve the accuracy and robustness of the trajectory estimation.

REFERENCES

[1] R. Mur-Artal, J. M. M. Montiel, and J. D. Tardos, "Orb-slam: a versatile and accurate monocular slam system," *IEEE transactions on robotics*, vol. 31, no. 5, pp. 1147–1163, 2015.

[2] C. Forster, Z. Zhang, M. Gassner, M. Werlberger, and D. Scaramuzza, "Svo: Semidirect visual odometry for monocular and multicamera systems," *IEEE Transactions on Robotics*, vol. 33, no. 2, pp. 249–265, 2016.

[3] J. Engel, V. Koltun, and D. Cremers, "Direct sparse odometry," *IEEE transactions on pattern analysis and machine intelligence*, vol. 40, no. 3, pp. 611–625, 2017.

[4] J. Zhang and S. Singh, "Loam: Lidar odometry and mapping in real-time." in *Robotics: Science and Systems*, vol. 2, no. 9. Berkeley, CA, 2014, pp. 1–9.

[5] T. Shan and B. Englot, "Lego-loam: Lightweight and ground-optimized lidar odometry and mapping on variable terrain," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2018, pp. 4758–4765.

[6] 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 Robotics and Automation Letters*, vol. 7, no. 1, pp. 446–453, 2022.

[7] G. P. C. Júnior, A. M. Rezende, V. R. Miranda, R. Fernandes, H. Azpúrua, A. A. Neto, G. Pessin, and G. M. Freitas, "Ekf-loam: An adaptive fusion of lidar slam with wheel odometry and inertial data for confined spaces with few geometric features," *IEEE Transactions on Automation Science and Engineering*, 2022.

[8] W. Hess, D. Kohler, H. Rapp, and D. Andor, "Real-time loop closure in 2d lidar slam," in *2016 IEEE international conference on robotics and automation (ICRA)*. IEEE, 2016, pp. 1271–1278.

[9] Y. Su, T. Wang, S. Shao, C. Yao, and Z. Wang, "Gr-loam: Lidar-based sensor fusion slam for ground robots on complex terrain," *Robotics and Autonomous Systems*, vol. 140, p. 103759, 2021.

[10] L. Xiong, R. Kang, J. Zhao, P. Zhang, M. Xu, R. Ju, C. Ye, and T. Feng, "G-vido: A vehicle dynamics and intermittent gnss-aided visual-inertial state estimator for autonomous driving," *IEEE Transactions on Intelligent Transportation Systems*, 2021.

[11] G. Cioffi, T. Cieslewski, and D. Scaramuzza, "Continuous-time vs. discrete-time vision-based slam: A comparative study," *IEEE Robotics and Automation Letters*, vol. 7, no. 2, pp. 2399–2406, 2022.

[12] X. Zuo, P. Geneva, W. Lee, Y. Liu, and G. Huang, "Lic-fusion: Lidar-inertial-camera odometry," in *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2019, pp. 5848–5854.

[13] W. Zhen, S. Zeng, and S. Soberer, "Robust localization and localizability estimation with a rotating laser scanner," in *2017 IEEE international conference on robotics and automation (ICRA)*. IEEE, 2017, pp. 6240–6245.

[14] H. Ye, Y. Chen, and M. Liu, "Tightly coupled 3d lidar inertial odometry and mapping," in *2019 International Conference on Robotics and Automation (ICRA)*. IEEE, 2019, pp. 3144–3150.

[15] 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 *2020 IEEE/RSJ international conference on intelligent robots and systems (IROS)*. IEEE, 2020, pp. 5135–5142.

[16] W. Xu, Y. Cai, D. He, J. Lin, and F. Zhang, "Fast-lio2: Fast direct lidar-inertial odometry," *IEEE Transactions on Robotics*, 2022.

[17] Z. Yu, L. Zhu, and G. Lu, "Vins-motion: Tightly-coupled fusion of vins and motion constraint," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2021, pp. 7672–7678.

[18] K. Huang, Y. Wang, and L. Kneip, "B-splines for purely vision-based localization and mapping on non-holonomic ground vehicles," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2021, pp. 5374–5380.

[19] H. Alismail, L. D. Baker, and B. Browning, "Continuous trajectory estimation for 3d slam from actuated lidar," in *2014 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2014, pp. 6096–6101.

[20] D. Droeschel and S. Behnke, "Efficient continuous-time slam for 3d lidar-based online mapping," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2018, pp. 5000–5007.

[21] Y. Cong, C. Chen, B. Yang, J. Li, W. Wu, Y. Li, and Y. Yang, "3d-cstm: A 3d continuous spatio-temporal mapping method," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 186, pp. 232–245, 2022.

[22] J. Lv, K. Hu, J. Xu, Y. Liu, X. Ma, and X. Zuo, "Clins: Continuous-time trajectory estimation for lidar-inertial system," in *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2021, pp. 6657–6663.

[23] C. Park, P. Moghadam, J. L. Williams, S. Kim, S. Sridharan, and C. Fookes, "Elasticity meets continuous-time: Map-centric dense 3d lidar slam," *IEEE Transactions on Robotics*, vol. 38, no. 2, pp. 978–997, 2021.

[24] A. Haarbach, T. Birdal, and S. Ilic, "Survey of higher order rigid body motion interpolation methods for keyframe animation and continuous-time trajectory estimation," in *2018 International Conference on 3D Vision (3DV)*. IEEE, 2018, pp. 381–389.

[25] H. Ovrén and P.-E. Forssén, "Trajectory representation and landmark projection for continuous-time structure from motion," *The International Journal of Robotics Research*, vol. 38, no. 6, pp. 686–701, 2019.

[26] C. Sommer, V. Usenko, D. Schubert, N. Demmel, and D. Cremers, "Efficient derivative computation for cumulative b-splines on lie groups," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 11 148–11 156.

[27] D. E. Smith and J. M. Starkey, "Effects of model complexity on the performance of automated vehicle steering controllers: Model development, validation and comparison," *Vehicle system dynamics*, vol. 24, no. 2, pp. 163–181, 1995.

[28] X. Zuo, M. Zhang, Y. Chen, G. Huang, Y. Liu, and M. Li, "Visual-based lifelong kinematics and pose estimation for skid-steering robots," *arXiv e-prints*, pp. arXiv–2006, 2020.

[29] M. Li and A. I. Mourikis, "Online temporal calibration for camera–imu systems: Theory and algorithms," *The International Journal of Robotics Research*, vol. 33, no. 7, pp. 947–964, 2014.

[30] S. Agarwal, K. Mierle, and T. C. S. Team, "Ceres Solver," 3 2022. [Online]. Available: https://github.com/ceres-solver/ceres-solver