Time Delay Error Online Correction of LiDAR-IMU System Through MSCKF Integrated DLRNN Method

Wanli Liu, Zhixiong Li, *Senior Member, IEEE*, Weihua Li, Paolo Gardoni, Haiping Du, *Senior Member, IEEE*, and Miguel Angel Sotelo, *Fellow, IEEE*

***Abstract*—When fusing the measurement data with different sampling frequencies from the light detection and ranging (LiDAR) and inertial measurement unit (IMU), their timestamps should be exactly aligned. However, in reality the timestamps of LiDAR and IMU are typically subject to different influences, which will inevitably generate the time delay error to reduce the accuracy and robustness of the LiDAR-IMU system. To this avail, this article proposes a new method that integrates the double layer recurrent neural network (DLRNN) and multistate constrained Kalman filter (MSCKF) to online correct the LiDAR-IMU time delay errors. In this new method, the MSCKF can improve the DLRNN training accuracy while in return the DLRNN can enhance the error estimating performance of the MSCKF. With this mutual improvement strategy, the time delay error can be precisely corrected in both the static and dynamic operation modes of the LiDAR-IMU system. The main contributions include: 1) Dual-information fusion is achieved between the DLRNN and MSCKF for accurate correction of the LiDAR-IMU time delay error; and 2) the proposed approach significantly improves the efficiency and accuracy of the time delay error correction in a real-time manner. Several experiments were carried out to evaluate the online correction performance of the proposed method. The experimental results demonstrate that the LiDAR-IMU time delay error can be accurately and quickly corrected, regardless the time-varying and unknown time delay. As a result, the positioning and navigation performance of the LiDAR-IMU system can be improved more appropriately for practical applications.**

***Index Terms*—Light detection and ranging inertial measurement unit (LiDAR-IMU), online correction, time delay error.**

I. Introduction

In recent years, the light detection and ranging inertial measurement unit (LiDAR-IMU) integration system has become a popular and effective choice for autonomous robots, vehicles, and airplanes to pursue navigation and positioning tasks; especially in the global positioning system (GPS) absence environments. The particular advantages of the LiDAR-IMU system lie in that the LiDAR can provide rich environmental information to effectively reduce the time drift errors of the IMU; and meanwhile, the IMU can render the object's angular and acceleration velocities to provide accurate motion estimation for the LiDAR [1], [2], [3].

Basically, fusing the measurement data from the LiDAR and IMU can solve the inertial navigation divergence to achieve high-precision positioning and navigation tasks. It is well known that the measuring rates of the LiDAR and IMU are usually different; in order to integrate the LiDAR and IMU measurements, their timestamps must be exactly known. Therefore, a timestamp for each measurement should be provided either from a sensor or a data control center [4], [5]. However, in static situation the timestamps of the LiDAR and IMU typically suffer from the triggering delay, transmission delay, or unsynchronized clock, resulting in a temporal misalignment (i.e., time delay) in the LiDAR-IMU data streams. Moreover, in dynamic situation, the time delay error will also be induced by the platform vibration, acceleration/deceleration motions, and sensor dynamic drift. These time delay errors will dramatically reduce the accuracy and robustness of the LiDAR-IMU system. For example, a few tens of milliseconds of the time delay error will result in severe misalignment between the LiDAR and IMU measuring results.

Manuscript received 12 June 2022; revised 24 January 2023, 28 May 2023, and 24 July 2023; accepted 11 September 2023. Date of publication 11 October 2023; date of current version 18 June 2024. Recommended by Technical Editor S. Jeon and Senior Editor Q. Zou. This work was supported in part by the Industry and Information Technology Development Program, China under Grant TC220A04W-1-167, in part by the National Natural Science Foundation of China under Grant 52274161 and Grant 51974290, in part by the GRAS Project of the Opole University of Technology, Poland, and in part by Norwegian Financial Mechanism 2014-2021 under Project Contract 2020/37/K/ST8/02748. *(Corresponding author: Zhixiong Li.)*

Wanli Liu is with the School of Mechanical and Electrical Engineering, China University of Mining and Technology, Xuzhou 221116, China (e-mail: 4830@cumt.edu.cn).

Zhixiong Li is with the Faculty of Mechanical Engineering, Opole University of Technology, 45-758 Opole, Poland (e-mail: zhixiong.li@yonsei.ac.kr).

Weihua Li and Haiping Du are with the School of Mechanical, Materials, Mechatronic, and Biomedical Engineering, University of Wollongong, Wollongong, NSW 2522, Australia (e-mail: weihuali@uow.edu.au; hdu@uow.edu.au).

Paolo Gardoni is with the Department of Civil and Environmental Engineering, University of Illinois Urbana-Champaign, Champaign, IL 61820 USA (e-mail: gardoni@illinois.edu).

Miguel Angel Sotelo is with the Department of Computer Engineering, University of Alcala, 28801 Madrid, Spain (e-mail: miguel.sotelo@uah.es).

Color versions of one or more figures in this article are available at https://doi.org/10.1109/TMECH.2023.3317018.

Digital Object Identifier 10.1109/TMECH.2023.3317018

Thus, it is a mandatory requirement for the LiDAR-IMU system to correct the time delay error [6], [7], [8].

Usually, the time delay error correction methods can be divided into two categories: one is hardware correction method and the other is optimization-based correction method. As for the hardware correction, Patil et al. [9] and Du et al. [10] developed a LiDAR-IMU fusion system to estimate real-time onboard time delay state of an unmanned aerial vehicle by using the hardware correction for time delay error. However, in this method specialized hardware is required, which increases the cost of the LiDAR-IMU system. Especially, the hardware correction method requires hardware synchronization, which is usually unavailable for majority consumers and self-assembled LiDAR-IMU suite. In contrast, the optimization-based correction method can avoid these drawbacks and has become an effective solution for time delay error correction [11].

The optimization-based correction method mainly includes the registration-based and filter-based correction approaches. The registration-based correction approaches use the overlapping field-of-views of multiple sensors to aligning their timestamps to estimate the time delay error. For example, Wu et al. [12] proposed a novel correspondence matching method for the hand-eye calibration system based on the neat Lie algebra formulation to find the well-matched relative time delay between sensors. Kelly et al. [13] aligned the curves in the three-dimensional (3-D) space between the inertial and visual sensors by converting the time delay calibration as a registration problem. Liu and Meng [14] proposed an online temporal calibration based on modified projection model for visual-inertial odometry. The filter-based correction approaches apply different filters or neural networks to correct the time delay errors. Niu et al. [15] introduced an online extended Kalman filter (EKF) to solve real-time synchronous issue in a LiDAR-aided INS for indoor mobile mapping. Liu [16] studied a hybrid method based on the iterative closest point and iterated sigma point Kalman filter to calibrate the time delay of a LiDAR-IMU system. Li et al. [17] used an unscented Kalman filter (UKF) to fuse multiple dissimilar sensors for temporal misalignment adjustment. Kumar et al. [18] employed the Kalman filter (KF) to estimate the time offset of a LiDAR-IMU based INS. Taylor et al. [19] presented a motion-based EKF to calibrate the extrinsic parameters and timing offsets of multimodal sensors. Zhang et al. [20] developed a distributed adaptive 3-D formation control method based on an improved recurrent neural network (RNN) to estimate the time-delay of a nonlinear multiagent system. Ma et al. [21] investigated a radial basis function neural network (RBFNN) distributed control method for nonlinear multiagent systems with time delays and external noises. Song et al. [22] proposed a resilience method for state estimation using discrete neural networks subject to multiple missing measurements and mixed time delays.

Literature review demonstrates that although the existing hardware and optimization-based methods can correct the time delay for the LiDAR-IMU system, the follows limitations still remain [23], [24], [25], [26].

1) In [9], [10], and [11], the proposed hardware correction approaches are not universal enough. Because of the assumption of a very little time delay, these approaches are unavailable for many practical applications where the time delay is varying over a long-term time. Moreover, existing hardware approaches often adopt offline estimation, an efficient online solution for time delay correction is still a challenge due to the triggering delay, transmission delay, unsynchronized clocks, and platform vibration.
2) In [12], [13], and [14], the proposed registration-based approaches have assumed that the time delay error is a fixed constant in the LiDAR-IMU system, while in practice the time delay error is always time-varying and unknown. Consequently, these approaches do not consider the uncertainties of the time delay in practical applications.
3) In [15], [16], [17], [18], and [19], the proposed filter-based approaches are applied to correct the time delay error for the LiDAR-IMU system. However, due to the nonlinear characteristics of the LiDAR and IMU, in GPS-absence environments, it will usually cause very large accumulative time delay errors. To effectively eliminate these errors, the filter-based approaches always need to resort the GPS assistance to realize a real-time loop closure correction; but the resorting is difficult/inefficient in the GPS-absence environments. In [20], [21], and [22], the neural network-based approaches are used to estimate the time delay error, but the training procedure of the neural networks cannot effectively eliminate the impact of zero-mean white Gaussian noise in a dynamic measurement scenario.

Based on the aforementioned limitations, a crucial problem is that the current filter-based or neural network-based approaches cannot effectively eliminate the time delay error for the LiDAR-IMU system. Usually, they merely add a neural network or a filter to train/estimate the time delay error without considering of optimizing the training parameters or the filter structures. To address these limitations with full consideration of computational accuracy, efficiency, and measurability in time delay correction, a novel method is proposed in this work by appropriately integrating the double layer recurrent neural network (DLRNN) and multistate constrained Kalman filter (MSCKF) for online correction of the LiDAR-IMU system. Generally, the current commonly used neural networks, including the feedforward neural networks (FNN), convolutional neural networks (CNN), and RNN, exist some limitations. For example, the FNN has a simple feedforward structure that cannot accurately record the time-domain behavior of a nonlinear system; the CNN has excellent performance in computer vision and natural language processing, but it is difficult to perform accurate estimation for time delay error in a dynamic system; the RNN only keep short-time memory of a dynamic system. To address these limitations, the DLRNN combines the advantages of FNN, CNN, and RNN using double hidden layers to realize more complex function fitting, which can be used to improve the computational efficiency and enhance response speed for LiDAR-IMU system.

Although the DLRNN possesses high training capability because of its advanced network structure, the training accuracy is insufficient for the time delay estimation in the LiDAR-IMU system in a real-time manner due to a large amount of structure

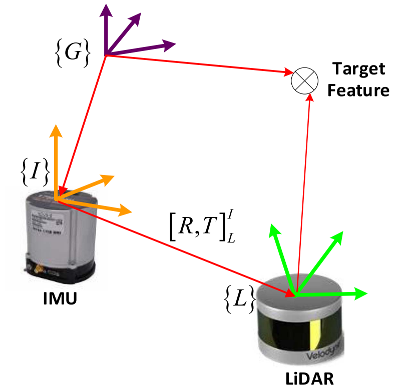

Fig. 1. Relationships between the $\{L\}$,$\{I\}$, and $\{G\}$ frames.

parameters that needed to optimize. The MSCKF is able to find the solution to a complex optimization problem, which can enhance the DLRNN training accuracy in a reasonable time. However, very little work has been done to develop the DLRNN-MSCKF solution for the time delay online correction. In order to bridge this research gap, a new DLRNN-MSCKF method is developed to improve the time delay error correction accuracy. The main contributions of this study can be summarized as follows [27], [28].

1) A new DLRNN-MSCKF method for time delay error online correcting is presented. In this method, the MSCKF can improve the DLRNN training accuracy while in return the DLRNN can enhance the estimating accuracy of the MSCKF. Furthermore, the presented method is capable to accurately estimate the time delay error in static and dynamic situations of the LiDAR-IMU system.
2) A Lagrange parabolic interpolation-based LiDAR measurement model is developed to accurately estimate the virtual interpolated values of the IMU attitudes, which does not require solving linear equations and can effectively reduce the rounding error for a large amount of data.

The presented MSCKF integrated DLRNN time delay error online correction method for the LiDAR-IMU system is evaluated by a series of experiments, and the results show that the time delay error can be accurately corrected in a real-time manner. Moreover, the presented DLRNN-MSCKF method is proved be effective to improve the positioning and navigation overall performance of the LiDAR-IMU system.

II. Time Delay Error Modelling

A. Coordinate Relationship Between LiDAR and IMU

As shown in Fig. 1, generally, the basic coordinate frames of the LiDAR-IMU include the LiDAR frame $\{L\}$, the IMU frame$\{I\}$, and the Global frame $\{G\}$. Assuming that a feature point ${}^{G}P$ at the frame $\{G\}$ corresponds to a point ${}^{L}P$ at the frame $\{L\}$, the position of ${}^{G}P$ in $\{L\}$ is calculated as [2]

$$ {}^{L}P = {}^{L}_{G}R \cdot {}^{G}P + {}^{L}_{G}T \tag{1} $$

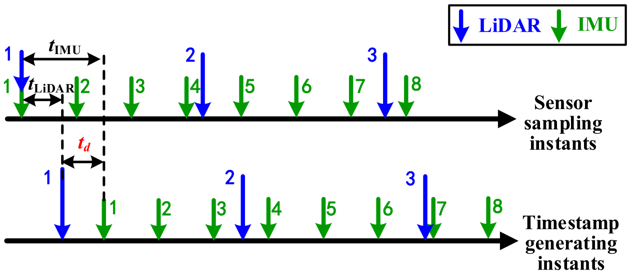

Fig. 2. Illustration of time delay between LiDAR and IMU.

where ${}^{L}_{G}T$ and ${}^{L}_{G}R$ denote the translation matrix and the rotation matrix between $\{G\}$ and $\{L\}$, respectively.

Normally, the IMU is consisted of three accelerometers and three gyroscopes. The gyroscopes measure the changing of the Euler angles and the accelerometers record the specific forces of the IMU. The IMU position information can be calculated from these angles and forces. The position relationship of the point P in $\{I\}$ and $\{G\}$ corresponding to $\{L\}$ can be described as

$$ {}^{I}P = {}^{I}_{L}R \cdot {}^{L}P + {}^{I}_{L}T \text{ and } {}^{G}P = {}^{G}_{I}R \cdot {}^{I}P + {}^{G}_{I}T \tag{2} $$

where ${}^{I}P$ is the point in $\{I\}$; ${}^{G}_{I}R$ and ${}^{G}_{I}T$ are the rotation matrix and translation matrix from $\{I\}$ to $\{G\}$, respectively.

Substituting (2) into (1), it yields

$$ {}^{L}P = {}^{L}_{G}R({}^{G}_{I}R \cdot {}^{I}P + {}^{G}_{I}T) + {}^{G}_{L}T = {}^{I}_{L}R \cdot {}^{I}P + {}^{I}_{L}T \tag{3} $$

where ${}^{I}_{L}T$and ${}^{I}_{L}R$ denote the translation matrix and rotation matrix from $\{L\}$ to $\{I\}$, respectively.

Equation (3) represents the coordinate transformation relationships between the LiDAR and IMU.

B. Time Delay Error Model

The time delay between the LiDAR and IMU is illustrated in Fig. 2. The upper plot denotes the sensor sampling time instant, and the lower plot represents the timestamping instant. The blue and green arrows, respectively, denote the LiDAR and IMU data sequences. Due to unequal sampling time, a time delay t_d is caused between LiDAR and IMU, which is described as [27]

$$ t_d = t_{\mathrm{IMU}} - t_{\mathrm{LiDAR}} \tag{4} $$

where t_{IMU} represents the sampling latency of the IMU and t_{LiDAR} denotes the sampling latency of the LiDAR. As t_{LiDAR} and t_{IMU} are unequal, the receiving sampled data simultaneously contains the time delay t_d. If the LiDAR sequence has a longer latency than that of the IMU sequence, t_d is a negative value; otherwise, and t_d is a positive value.

Assuming that the LiDAR-IMU sensor can be regarded as a rigid body moving in the space, the parameterization representation can be used to model the time delay error for the LiDAR-IMU. The system parameterization state vector can be described as

$$ \begin{aligned} \mathrm{X} &= \begin{bmatrix} \mathrm{X}_I & \mathrm{X}_L & t_d \end{bmatrix}^T \\ &= \begin{bmatrix} {}^{G}Q_I & {}^{G}P_I & {}^{G}V_I & {}^{G}b_g & {}^{G}b_a & {}^{I}P_L & {}^{I}Q_L & t_d \end{bmatrix}^T \end{aligned} \tag{5} $$

where X is the LiDAR-IMU state vector, X_I and X_L are IMU and LiDAR state vector; ${}^{G}P_I$ and ${}^{G}V_I$ denote the position and linear velocity of the IMU at $\{G\}$; ${}^{G}Q_I$ represents the unit quaternion orientation with the rotation ${}^{G}_{I}R$ from $\{I\}$ to $\{G\}$; ${}^{G}b_g$ and ${}^{G}b_a$ denote the bias vectors of the gyroscope and accelerometer of the IMU at $\{G\}$, respectively; ${}^{I}P_L$ represents the position of the LiDAR at $\{I\}$; ${}^{I}Q_L$ denotes the unit quaternion orientation with the rotation ${}^{I}_{L}R$ from $\{L\}$ to $\{I\}$.

C. IMU Measurement Model

The measurement model of the IMU accompanied with the time delay t_d can be calculated as

$$\mathrm{X}_I = \begin{bmatrix} {}^{G}Q_I & {}^{G}P_I & {}^{G}V_I & {}^{G}b_g & {}^{G}b_a & t_d \end{bmatrix}^T. \tag{6}$$

Basically, the IMU measures the rotating motions of three orthogonal axes, and the gyroscopes and accelerometers provide the rotational velocity and linear acceleration, respectively.

The linearized continuous time measurement model for $\tilde{\mathrm{X}}_I(t)$ can be written as

$$\tilde{\mathrm{X}}_I\,(t) = A \cdot \mathrm{X}_I + B \cdot n_I \tag{7}$$

where $\tilde{\mathrm{X}}_I(t)$ denotes the IMU measurement model for X_I; $n_I = [n_g \quad n_{\omega g} \quad n_a \quad n_{\omega a}]^T$ represents the continuous-time IMU noise with its covariance matrix as $diag\{\sigma_g^2, \sigma_{\omega g}^2, \sigma_a^2, \sigma_{\omega a}^2\}$. The detailed expressions of A and B have been given in the Appendix.

D. Interpolation-Based LiDAR Measurement Model

Similarly, the state vector of the LiDAR augmented with the time delay t_d can be calculated as

$$\mathrm{X}_{L_k}(t) = \begin{bmatrix} {}^{I}P_{L_k}(t) & {}^{I}Q_{L_k}(t) & t + t_d \end{bmatrix}^T \tag{8}$$

where X_{L_k} is the LiDAR state vector at timestep k; superscript I denotes the $\{I\}$ frame. If define $t = t_{\mathrm{IMU}}$ as the current timestamped of the LiDAR-IMU, the sampling time of the LiDAR with time delay t_d can be denoted as $t + t_d$.

Due to the sampling frequency of the LiDAR is much smaller than the IMU, it also means the sampling timestamp of LiDAR sensor much larger than the IMU sensor. Note that the LiDAR timestamp refers to each scan point timestamp, which can be calculate through the emission timestamp of laser pulse generator and the scanning frequency of the LiDAR [8]. Thus, it is necessary to interpolate the collected LiDAR scanning data before the time synchronization between the LiDAR and IMU data stream. Generally, the current commonly used interpolation methods include the linear interpolation, Newton interpolation, Hermite interpolation, and Lagrange parabolic interpolation. The advantage of the Lagrange parabolic interpolation is that it does not require to solve the linear equations, also can effectively reduce the rounding error for a large amount of data; moreover, the calculation effect is much better than other interpolation methods. Therefore, the Lagrange parabolic interpolation method [29] is adopted to estimate LiDAR measuring position and attitude [30], [31].

At time step k between the temporally neighboring $2n + 1$ IMU positions and attitudes, the LiDAR measuring position, and attitude can be expressed as

$$\begin{cases} {}^{G}\bar{P}_{L_k}(t) = \sum_{i=k-n}^{k+n} \lambda_i {}^{G}P_{I_i} \left(\prod_{j=k-n}^{k+n} \frac{t_i - t_j}{t - t_j} \right) \\ {}^{G}\bar{Q}_{L_k}(t) = \sum_{i=k-n}^{k+n} \lambda_i {}^{G}Q_{I_i} \left(\prod_{j=k-n}^{k+n} \frac{t_i - t_j}{t - t_j} \right) \\ \lambda_i = \left(t_{d_k} + t_{F_k} t_r \right) / t_{I_k} \end{cases} \tag{9}$$

where ${}^{G}\bar{P}_{L_k}$ and ${}^{G}\bar{Q}_{L_k}$ denote the interpolation-based position and attitude of the LiDAR measurements in the $\{G\}$ frame, respectively; λ_i represents the interpolation ratios determined according to the time delay t_{d_k} between the LiDAR $({}^{G}P_{L_k}, {}^{G}Q_{L_k})$ and IMU $({}^{G}P_{I_k}, {}^{G}Q_{I_k})$; $t \in [t_{k-n}, t_{k+n}]$, t_r is the LiDAR rolling scanning time, t_{I_k} is the time interval between the IMU measurements X_{I_k} and $\mathrm{X}_{I_{k+1}}$; t_{F_k} is the time of the target feature F_i sampled by the LiDAR.

The least square optimization is employed to minimize the summation of cost terms arising from (9). The minimization problem can be expressed as

$$\begin{cases} \xi_{P_{L_k}} = \underset{P_{L_k}}{\arg\min} \left\| {}^{G}\bar{P}_{L_k}(t) \right. \\ \qquad \left. - \sum_{i=k-n}^{k+n} \lambda_i {}^{G}P_{I_i} \left(\prod_{j=k-n}^{k+n} \frac{t_i - t_j}{t - t_j} \right) \right\| \\ \xi_{Q_{L_k}} = \underset{Q_{L_k}}{\arg\min} \left\| {}^{G}\bar{Q}_{L_k}(t) \right. \\ \qquad \left. - \sum_{i=k-n}^{k+n} \lambda_i {}^{G}Q_{I_i} \left(\prod_{j=k-n}^{k+n} \frac{t_i - t_j}{t - t_j} \right) \right\| \end{cases} \tag{10}$$

where $\xi_{P_{L_k}}$ and $\xi_{Q_{L_k}}$ are the approximation errors of the position and attitude of the LiDAR measurements, which can be minimized using a sufficient number of interpolation iterations. Considering the time delay t_d, the estimated position and attitude, ${}^{G}\bar{P}_{L_k}(t)$ and ${}^{G}\bar{Q}_{L_k}(t)$, can be obtained using the adjacent positions and attitudes of the IMU by employing the interpolation-based constrains. Thus, (8) is rewritten as

$$\bar{\mathrm{X}}_{L_k}(t) = \begin{bmatrix} {}^{I}\bar{P}_{L_k}(t) & {}^{I}\bar{Q}_{L_k}(t) & t + t_d \end{bmatrix}^T \tag{11}$$

where $\bar{\mathrm{X}}_{L_k}$ denote the interpolation-based LiDAR state vector.

Suppose the *i*th target feature observed by the LiDAR as F_i, then, the geometric constraint describing the state vector $\bar{\mathrm{X}}_{L_k}^{F_i}$ of the *i*th target feature can be expressed as

$$\bar{\mathrm{X}}_{L_k}^{F_i}(t) = h\left({}^{L_k}P_{F_i}(t + t_d) \right) + n_{L_k}(t + t_d) \tag{12}$$

where $\bar{\mathrm{X}}_{L_k}^{F_i}$ demotes the state vector of F_i in $\{L\}$ frame at *k*th measurement; $h(\cdot)$ denotes the scanned equation of LiDAR, and $h(F_i) = [F_{x_i}, F_{y_i}, F_{z_i}]^T = [\rho_i \sin\alpha_i \cos\beta_i, \rho_i \cos\alpha_i \cos\beta_i, \rho_i \sin\beta_i]^T$, where ρ denotes the measuring distance of the LiDAR, α and β, respectively, represent the horizontal and vertical angles of the LiDAR; n_{L_k} represents the measuring noise vector, which is modelled as a zero-mean Gaussian covariance matrix $diag\{\sigma_\rho^2, \sigma_\alpha^2, \sigma_\beta^2\}$; ${}^{L_k}P_{F_i}$ denotes the position state model of the featureF_i in the $\{L\}$ frame, which also can be calculated by the coordinate translation process as

$${}^{L_k}P_{F_i}(t + t_d) = {}^{L}_{I}R^{I}_{G}R(t + t_d)\left({}^{G}P_{F_i} - {}^{G}P_I(t + t_d) \right) + {}^{L}P_I \tag{13}$$

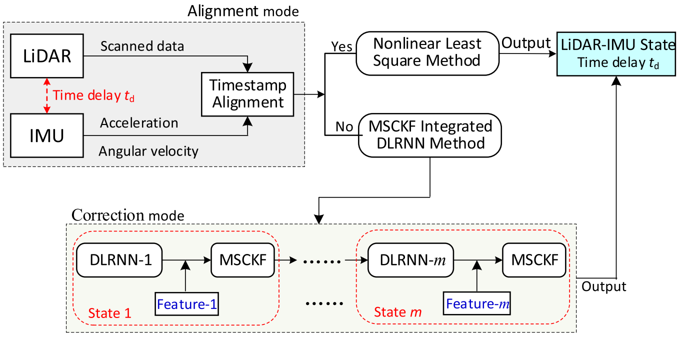

Fig. 3. Basic structure of the proposed method.

where ${}^G P_{F_i}$ denotes the position of the target feature F_i at the $\{G\}$ frame; ${}^G P_I(t+t_d)$ represents the position of the IMU with regard to the $\{G\}$ frame at the time $t+t_d$; ${}^L P_I$ represents the known position of the IMU in regard to the $\{L\}$ frame.

Based on (13), if the target feature position F_i at the $\{G\}$ frame is known as *a priori*, the continuous time measurement model for $\bar{\mathrm{X}}^{F_i}_{L_k}(t)$ can be expressed as

$$\widetilde{\bar{\mathrm{X}}}^{F_i}_{L_k}(t) = h\left({}^L_I R^I_G R(t+t_d)\left({}^G P_{F_i} - {}^G P_I(t+t_d)\right) + {}^L P_I\right) + n_{L_k}(t+t_d). \tag{14}$$

Equation (14) represents the interpolation-based LiDAR measurement model with the target features.

E. Optimization of Time Delay

According to the established measurement models, the time delay correction function can be converted into typical LiDAR-IMU optimization-based frameworks, where the time delay error correction can be regarded as a nonlinear optimization problem. Specifically, the optimization of the time delay can be augmented through the LiDAR-IMU state variables, and the time delay error can be calculated through minimizing the nonlinear least square problems as follows:

$$\Theta_{t_d} = \operatorname{argmin} \sum_{i=1}^{n} \left\| \tilde{\mathrm{X}}_I(t_i)\,|F_i - \tilde{\tilde{\mathrm{X}}}_L(t_i)\,|F_i \right\|^2 \tag{15}$$

where Θ_{t_d}donates the optimal value of t_d,$\tilde{X}_I(t_i)$ and $\tilde{\tilde{\mathrm{X}}}_L(t_i)$ represent the measurement model in time t_i of the IMU and LiDAR, also their expression have been given in (7) and (14), $F_i$represents the ith target feature.

III. MSCKF Integrated DLRNN Method

Due to different sampling rates, one should correct the time delay error in the LiDAR-IMU. A new MSCKF integrated DL-RNN method is proposed to minimize the time delay error. The basic structure of the proposed method is shown in Fig. 3, which mainly includes two modes: 1) timestamp alignment mode and 2) time delay error correction mode.

At the beginning, in the timestamp alignment mode, we use the dual quaternion algorithm [16] to calculate the IMU position, velocity and angle information and adopt the Extended Kalman filter method [18] to eliminate the IMU drift noise, and then combine the LiDAR scanned point data to achieve the coordinates alignment for the LiDAR-IMU system. Also, the timestamps of LiDAR and IMU sensors will be aligned through system initialization. Subsequently, at a given time, the timestamp alignment mode switch condition is determined by the timestamps recorded by the LiDAR and IMU sensors, including the IMU measurements timestamps and the LiDAR scanning point data timestamps. Compared with the timestamp of the LiDAR and IMU sensors, if the timestamp is aligned between the LiDAR and IMU sensors (i.e., the alignment error is less than the design value τ), the switch mode will turn to the nonlinear least square method but not to the correction mode to directly output the time delay error. Otherwise, if the timestamp of LiDAR and IMU sensors is not aligned (i.e., the alignment error is more than the τ), the switch mode will turn ON the correction mode to estimate the time delay t_d. The time delay error correction mode mainly includes the design of DLRNN and MSCKF, MSCKF integrated DLRNN. During the t_d calculation, the current states of the LiDAR-IMU measurements as well as the previous time-related state i $(1 \leq i \leq m)$ will be used to train a LiDAR-IMU state model to estimate the time delay error. The value of m is determined by the DLRNN training accuracy and MSCKF estimating accuracy.

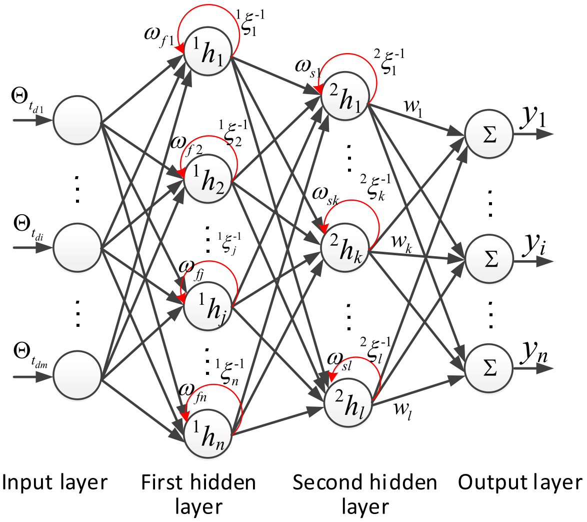

Fig. 4. Architecture of DLRNN.

A. Design of DLRNN

Because the time delay function of the LiDAR-IMU system is quite complex, the standard NN with a single layer is hard to approximate it. In addition, the standard NN may need a large number of neurons for training, which will result in serious computational complexity, long training time, and high memory consumption. To address these limitations, a DLRNN is developed as Fig. 4, which is a dynamic nonlinear multilayer perceptron trained with the specialized weight adaptation algorithm. Its unique feature distinguished from the traditional neural networks is the two-hidden-layer feedback loop in the topology, which enable the DLRNN to dynamically store the memories of previous states to achieve nonlinear estimation. Moreover, the double layer activation structure makes the DLRNN not only deliver more information in the input layer but also transmitting more information to the output layer, which makes it achieve better approximation performance for time-varying systems.

As shown in Fig. 4, the DLRNN consists of one input layer, two hidden layers with the feedback units and one output layer. The periodic feedback loop with specialized weight is embedded in each hidden layer node, where the past weight and output information will be stored and treated as the feedback. Compared with the standard NN with fixed widths and unchanged depth, the training parameters of the DLRNN can be updated and eventually stabilized at the optimal values. The detailed explanation of the DLRNN is given as follows.

1) Input layer: The main function of the input layer is to complete the transmission of the inputs, according to (15). The inputs of the DLRNN are $\mathrm{X} = [\Theta_{\mathrm{td1}}, \Theta_{\mathrm{td2}}, \ldots, \Theta_{\mathrm{tdi}}, \cdots \Theta_{\mathrm{tdm}}]^{\mathrm{T}}\ i = 1, 2, \ldots, m$, where $\Theta_{t_{di}}$ denotes the *i*th node input.

2) Hidden layer: This layer is regarded as the activation functions to train the outputs from the input layer. Basically, each hidden layer usually selects the Gaussian function as the activation function. The first hidden layer Gaussian function can be computed as

$$\begin{cases} {}^{1}h_j = e^{-{}^{1}net_j} \\ {}^{1}net_j = \sum_{i=1}^{m} \frac{\left(x_i + \omega_{fj} \cdot {}^{1}\xi_j^{-1} - {}^{1}c_{ij}\right)}{{}^{1}b_{ij}^{2}} \end{cases} \tag{16}$$

where superscript indicates the layer number and subscript indicates the node number; ${}^{1}h_j, j = 1, 2, \ldots, n$ denotes the Gaussian vector in the first hidden layer; ${}^{1}net_j$ represents the network output in *j*th node; $e^{-{}^{1}net_j}$ denotes the input of in the first layer; $B = [b_{11}, \ldots, b_{1n}, b_{21}, \ldots, b_{2n}, \ldots, b_{mn}]^T$,$C = [c_{11}, \ldots, c_{1n}, c_{21}, \ldots, c_{2n}, \ldots, c_{mn}]^T$; b_{ij} and c_{ij}, respectively, represent the width and center vector of the Gaussian function in the *j*th node for the *i*th input; $x_i = \Theta_{t_{di}}$ the *i*th node input; ${}^{1}\xi_j^{-1}$ is the feedback value in the *j*th node; $W_f = [\omega_{f1}, \omega_{f2}, \ldots, \omega_{fn}]$, ω_{fj} denote the weight for the *j*th node.

Similarly, the Gaussian function for the second hidden layer is calculated as

$$\begin{cases} {}^{2}h_k = e^{-{}^{2}net_k} \\ {}^{2}net_k = \sum_{i=1}^{n} \frac{\left(x_i + \omega_{sk} \cdot {}^{2}\xi_k^{-1} - {}^{2}c_{ik}\right)}{{}^{2}b_{ik}^{2}} \end{cases} \tag{17}$$

where ${}^{2}h_j, j = 1, 2, \ldots, k$ denotes the Gaussian vector in the second hidden layer;$W_s = [\omega_{s1}, \omega_{s2}, \omega_{sk}, \ldots, \omega_{sl}]$, $k = 1, 2, \ldots, l$,ω_{sk} denotes the weight in the *k*th node at second hidden layer;$x_i = \Theta_{t_{di}}$ the *i*th node input; ${}^{2}\xi_k^{-1}$ is the feedback value in the *k*th node. The second hidden layer is to further optimize the nonlinear Gaussian function and to reduce the number of nodes in the network.

3) Output layer: Each node in the output layer connects to all nodes in the second hidden layer using weight $W = [w_1, w_2, \ldots, w_l]$. The output vector $\mathrm{Y} = [y_1, y_2, \ldots, y_n]^T$ can be described as

$$\mathrm{Y} = WH = w_1 h_1 + w_2 h_2 + \cdots + w_l h_l \tag{18}$$

where $H = [h_1, h_2, \ldots, h_l]$ is the Gaussian vector output from the second hidden layer.

Subsequently, according to the optimal approximation theory, the optimal values for the parameters W_f^*, W_s^*, W^*, B^*, and C^* can be applied to estimating the output vector $\mathbf{Y}$ as $\mathrm{Y} = W^* H^* + \varepsilon$, where $H^* = H^*(X, W_f^*, W_s^*, B^*, C^*)$, ε is the minimum error between the estimated and actual values. The approximation error vector between the actual value $\mathbf{Y}$ and estimated value $\hat{\mathrm{Y}} = \hat{W}\hat{H}(X, \hat{W}_f, \hat{W}_s, \hat{B}, \hat{C})$can be calculated as

$$\begin{aligned} Y - \widehat{Y} &= W^*\ H^* - \widehat{W}\widehat{H} + \varepsilon = W^* \left(\widehat{H} + \overset{\leftrightarrow}{H}\right) - \widehat{W}\widehat{H} + \varepsilon \\ &= W^*\ \widehat{H} + W^* \overset{\leftrightarrow}{H} - \widehat{W}\widehat{H} + \varepsilon = \overset{\leftrightarrow}{W}\ \widehat{H} + \widehat{W}\overset{\leftrightarrow}{H} + \overset{\leftrightarrow}{W}\overset{\leftrightarrow}{H} \\ &+ \varepsilon = \overset{\leftrightarrow}{W}\ \widehat{H} + \widehat{W}\overset{\leftrightarrow}{H} + \varepsilon_0 \end{aligned} \tag{19}$$

where $\overset{\leftrightarrow}{W} = W^* - \widehat{W}$ and $\overset{\leftrightarrow}{H} = H^* - \widehat{H}$, $\varepsilon_0 = \overset{\leftrightarrow}{W}\ \overset{\leftrightarrow}{H} + \varepsilon$.

Next, to obtain the adaptive law for the DLRNN parameters, the Taylor series expansion linearization technique is applied to the DLRNN activation function $\overset{\leftrightarrow}{H}$, which can be written as

$$\begin{aligned} \overset{\leftrightarrow}{H} &= \left.\frac{\partial \overset{\leftrightarrow}{H}}{\partial B}\right|_{B=\widehat{B}} \left(B^* - \widehat{B}\right) + \left.\frac{\partial \overset{\leftrightarrow}{H}}{\partial C}\right|_{C=\widehat{C}} \\ &\left(C^* - \widehat{C}\right) \left.\frac{\partial \overset{\leftrightarrow}{H}}{\partial W_f}\right|_{W_f=\widehat{W}_f} \left(W_f^* - \widehat{W}_f\right) + \left.\frac{\partial \overset{\leftrightarrow}{H}}{\partial W_s}\right|_{W_s=\widehat{W}_s} \\ &\times \left(W_s^* - \widehat{W}_s\right) + \sigma \\ &= DH_B\overset{\leftrightarrow}{B} + DH_C\overset{\leftrightarrow}{C} + DH_{W_f}\overset{\leftrightarrow}{W}_f + DH_{W_s}\overset{\leftrightarrow}{W}_s + \sigma \end{aligned} \tag{20}$$

where σ denotes a high order term; $\overset{\leftrightarrow}{B} = [\overset{\leftrightarrow}{b}_{11}, \ldots, \overset{\leftrightarrow}{b}_{1n}, \overset{\leftrightarrow}{b}_{21}, \ldots, \overset{\leftrightarrow}{b}_{2n}, \ldots, \overset{\leftrightarrow}{b}_{mn}]^T$, $\overset{\leftrightarrow}{C} = [\overset{\leftrightarrow}{c}_{11}, \ldots, \overset{\leftrightarrow}{c}_{1n}, \overset{\leftrightarrow}{c}_{21}, \ldots, \overset{\leftrightarrow}{c}_{2n}, \ldots, \overset{\leftrightarrow}{c}_{mn}]^T$, $\overset{\leftrightarrow}{W}_f{=}[\overset{\leftrightarrow}{\omega}_{f1}, \overset{\leftrightarrow}{\omega}_{f2}, \ldots, \overset{\leftrightarrow}{\omega}_{fn}]$, $\overset{\leftrightarrow}{W}_s{=}[\overset{\leftrightarrow}{\omega}_{s1}, \overset{\leftrightarrow}{\omega}_{s2}, \ldots, \overset{\leftrightarrow}{\omega}_{sl}]$, $DH_B = [\frac{\partial \overset{\leftrightarrow}{H}_1}{\partial B}\ \ \ldots\ \ \frac{\partial \overset{\leftrightarrow}{H}_l}{\partial B}]^T$, $D_{H_C}{=}[\frac{\partial \overset{\leftrightarrow}{H}_1}{\partial C}\ \ \ldots\ \ \frac{\partial \overset{\leftrightarrow}{H}_l}{\partial C}]^T$, $DH_f = [\frac{\partial \overset{\leftrightarrow}{H}_1}{\partial W_f}\ \ \ldots\ \ \frac{\partial \overset{\leftrightarrow}{H}_l}{\partial W_f}]^T$, $DH_s = [\frac{\partial \overset{\leftrightarrow}{H}_1}{\partial W_s}\ \ \ldots\ \ \frac{\partial \overset{\leftrightarrow}{H}_l}{\partial W_s}]^T$.

Substituting (20) into (19), we can obtain

$$\begin{aligned} Y - \widehat{Y} &= \overset{\leftrightarrow}{W}\ \widehat{H} + \widehat{W} \\ &\left(DH_B\overset{\leftrightarrow}{B} + DH_C\overset{\leftrightarrow}{C} + DH_{W_f}\overset{\leftrightarrow}{W}_f + DH_{W_s}\overset{\leftrightarrow}{W}_s + \sigma\right) + \varepsilon_0 \\ &= \overset{\leftrightarrow}{W}\ \widehat{H} + \hat{W} \\ &\left(DH_B\overset{\leftrightarrow}{B} + DH_C\overset{\leftrightarrow}{C} + DH_{W_f}\overset{\leftrightarrow}{W}_f + DH_{W_s}\overset{\leftrightarrow}{W}_s\right) + \sigma_0 \end{aligned} \tag{21}$$

where $\sigma_0 = \hat{W}\sigma + \varepsilon_0$ represents the lumped approximation error that is expected to be bounded; for example, $\sigma_0 \leq \sigma_d$ and σ_d is a small positive constant.

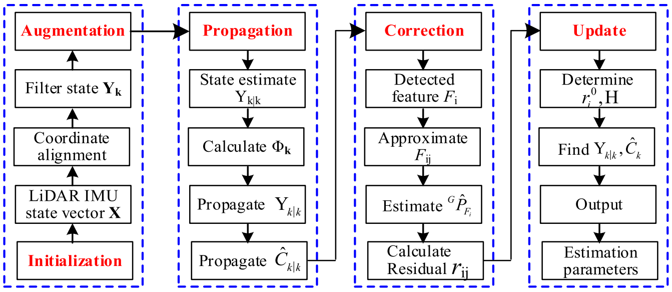

Fig. 5. Basic workflow of MSCKF.

Giving a certain value for ε and σ, based on (21) the optimal state vector $\mathbf{Y}$ can be obtained.

B. Design of MSCKF

In this section, the MSCKF algorithm is designed to estimates the time delay error between the LiDAR and IMU. Comparing with others filter algorithms such as the KF, ELF, UKF, the MSCKF has two advantages for accurate time delay estimation: time consistency and high accuracy constraints properties, which guarantee appropriate state constraints and more accurate estimations for time delay error. The basic workflow of the MSCKF is illustrated in Fig. 5, which mainly include the state processes of initialization, augmentation, propagation, correction, and update.

1) State Initialization: After the state vector X of the LiDAR-IMU is trained by the DLRNN, the optimal output vector Y is obtained. At the timestep *k*, the MSCKF error state vector can be expressed as

$$\tilde{Y}_k = \begin{bmatrix} \tilde{Y}_{I_k}^T & \tilde{Y}_{L_{k-1}}^T & \cdots & \tilde{Y}_{L_j}^T & \cdots & \tilde{Y}_{L_{k-N}}^T \end{bmatrix}^T \tag{22}$$

where $\tilde{Y}_k$ is the DLRNN trained output error vector at timestep k; $\tilde{Y}_{I_k}^T$represents the IMU error vector at timestep k; $Y_{L_j}^T (j = k - N, \ldots, k-1)$ denote the past *N* pieces of LiDAR observed data.

According to (22), in the LiDAR-IMU covariance propagation, the state covariance matrix $\hat{C}_k$ is written as

$$\hat{C}_k = \begin{bmatrix} \hat{C}_{II,k} & \hat{C}_{IL,k} \\ \hat{C}_{IL,k} & \hat{C}_{LL,k} \end{bmatrix} \tag{23}$$

where $\hat{C}_{II,k}$ denotes the current IMU 12×12 covariance state matrix at timestep k; $\hat{C}_{LL,k}$ represents the LiDAR $6N \times 6N$ covariance state matrix; $\hat{C}_{IL,k}$denotes $12 \times 6N$ cross covariance between the current the IMU states and the last LiDAR states.

2) State Augmentation: When a new LiDAR state becomes available, the MSCKF state should be augmented with the current LiDAR state. Based on (23), the LiDAR state and transformation $({}_{L}^{I}R, {}_{L}^{I}T)$ can be used to calculate the current IMU state. Suppose that the MSCKF state has already been augmented into N LiDAR states, when the $(N$+1) th LiDAR state is added to the LiDAR-IMU state vector, it yields

$$\begin{bmatrix} \tilde{Y}_k^T & {}^{I}P_{L_{k+1}} & {}^{I}Q_{L_{k+1}} \end{bmatrix}^T \rightarrow \tilde{Y}_{k+1}. \tag{24}$$

The corresponding augmented MSCKF state covariance is

$$\begin{bmatrix} \mathrm{I}_{12+6N} \\ J_k \end{bmatrix} \hat{C}_k \begin{bmatrix} \mathrm{I}_{12+6N} \\ J_k \end{bmatrix}^T \rightarrow \hat{C}_{k+1} \tag{25}$$

where J_k is the Jacobian matrix.

3) State Covariance Propagation: According to (23), the LiDAR state covariance and the IMU-LiDAR cross correlation can be calculated as

$$\begin{cases} \hat{C}_{II,k+1}^{-} = \hat{C}_{II,k} \\ \hat{C}_{IL,k+1}^{-} = \Phi\left(t_k + \Delta t, t_k\right) \hat{C}_{IL,k} \end{cases} \tag{26}$$

where Δt denotes the IMU sampling rate, the state transition matrix $\Phi(t_k + \Delta t, t_k)$, and the predicted IMU state covariance $\hat{C}_{II,k+1}^{-}$ can be expressed as

$$\begin{cases} \Phi\left(t_k + \Delta t, t_k\right) = \mathrm{I}_{12} + F\Delta t \\ \hat{C}_{II,k+1}^{-} = \Phi\left(t_k + \Delta t, t_k\right) \hat{C}_{II,k} \Phi^T\left(t_k + \Delta t, t_k\right) \\ \qquad + GQ_I G^T \Delta t \end{cases} \tag{27}$$

where Q_I represents the unit quaternion of the IMU.

4) Time Delay State Correction: When a target feature F_i is sampled N times by the LiDAR at timestep k, a set of observations of F_i in the $\{L\}$ frame can be expressed as ${}^{L_j}P_{F_i} = \{{}^{L_{k-N}}P_{F_i}, \ldots, {}^{L_{k-1}}P_{F_i}\}$. Based on the Gauss–Newton optimization method, the ${}^{L}P_{F_i}$ can be approximated as a set of multivariate Gaussian distributions with means $\mu = \{\mu_{k-N}, \ldots, \mu_{k-1}\}$and covariance $\upsilon = \{\upsilon_{k-N}, \ldots, \upsilon_{k-1}\}$. The best estimation of ${}^{G}\hat{P}_{F_i}$ in $\{G\}$ frame can be obtained using μ and υ.

Consider the jth observation of ${}^{L_j}P_{F_i}$ $(j = k - N, \ldots, k - 1)$ in the $\{L\}$frame, which is approximated as a multivariate Gaussian distribution with mean $\mu_j$and covariance υ_j. The feature position ${}^{L_j}P_{F_i}$ is related to the $\{G\}$ frame by

$${}^{L_j}P_{F_i} = {}_{I}^{L}R_{j,k-1} \cdot {}_{G}^{I}R_{j,k-1} \left({}^{G}P_{F_i} - {}^{G}P_{j,k-1}\right) + {}^{L}P_I. \tag{28}$$

The N observations of the feature $F_i$can be stacked as

$$\begin{bmatrix} {}^{L_{k-N}}P_{F_i} \\ \vdots \\ {}^{L_{k-1}}P_{F_i} \end{bmatrix} = \begin{bmatrix} {}_{I}^{L}R_{k-N,k-1} \cdot {}_{G}^{I}R_{k-N,k-1} \\ \vdots \\ {}_{I}^{L}R_{k-1,k-1} \cdot {}_{G}^{I}R_{k-1,k-1} \end{bmatrix} \cdot \left[{}^{G}P_{F_i}\right] + \begin{bmatrix} {}^{L}P_I - {}_{I}^{L}R_{k-N,k-1} \cdot {}_{G}^{I}R_{k-N,k-1} \cdot {}^{G}P_{k-N,k-1} \\ \vdots \\ {}^{L}P_I - {}_{I}^{L}R_{k-1,k-1} \cdot {}_{G}^{I}R_{k-1,k-1} \cdot {}^{G}P_{k-1,k-1} \end{bmatrix}. \tag{29}$$

The best estimation of ${}^{G}\hat{P}_{F_i}$ can be calculated using a weighted least square method as

$${}^{G}\hat{P}_{F_i} = \left(A^T \upsilon^{-1} A\right)^{-1} A^T \upsilon^{-1} \left(\mu \cdot {}^{L_j}P_{F_i} - B\right) \tag{30}$$

where $A = \begin{bmatrix} {}_{I}^{L}R_{k-N,k-1} \cdot {}_{G}^{I}R_{k-N,k-1} \\ \vdots \\ {}_{I}^{L}R_{k-1,k-1} \cdot {}_{G}^{I}R_{k-1,k-1} \end{bmatrix}$, $B = \begin{bmatrix} {}^{L}P_I - {}_{I}^{L}R_{k-N,k-1} \cdot {}_{G}^{I}R_{k-N,k-1} \cdot {}^{G}P_{k-N,k-1} \\ \vdots \\ {}^{L}P_I - {}_{I}^{L}R_{k-1,k-1} \cdot {}_{G}^{I}R_{k-1,k-1} \cdot {}^{G}P_{k-1,k-1} \end{bmatrix}$.

Since the state augmentation between the LiDAR and IMU contains the time delay t_d, the residual of the target feature between the actual and expected measurements is expressed as

$$r_{ij} = \overset{\approx}{\mathrm{X}}{}_{L_j}^{F_i} - h\left(\widetilde{{}^{L}Q}_{j|k-1}, \widetilde{{}_{L}^{I}R}_{j|k-1}, \widetilde{{}^{G}P}_{F_i}\right) \tag{31}$$

where r_{ij} is the residual of the target feature F_i in the LiDAR; $\widetilde{{}^{L}Q}_{j|k-1}$ is the current estimated error for the *j*th LiDAR measurement state; $\widetilde{{}_{L}^{I}R}_{j|k-1}$ is the *j*th LiDAR-to-IMU transformation error; $\overset{\approx}{\mathrm{X}}{}_{L_j}^{F_i}$ has been defined in (14).

The linearized approximation for (29) becomes as

$$\begin{aligned} r_{ij} \simeq{} & \mathrm{H}_{Q_{ij}} \cdot \widetilde{{}^{L}Q}_{j|k-1} + \mathrm{H}_{R_{ij}} \cdot \widetilde{{}_{L}^{I}R}_{j|k-1} \\ & + \mathrm{H}_{F_{ij}} \cdot \widetilde{{}^{G}P}_{F_i} + \mathrm{H}_{t_{ij}} \cdot \tilde{t}_d + n_{L_{ij}} \end{aligned} \tag{32}$$

where $H_{Q_{ij}}$ and $H_{F_{ij}}$ represent the corresponding Jacobians evaluated based on $\widetilde{{}^{L}Q}_{j|k-1}$and $\widetilde{{}^{G}P}_{F_i}$; $H_{R_{ij}}$denotes the measurement Jacobian in regard to the LiDAR-IMU state evaluated using $\widetilde{{}_{L}^{I}R}_{j|k-1}$; $H_{t_{ij}}$ is the Jacobian of $\tilde{t}_d$. These Jacobians matrix $H_{F_{ij}}$,$H_{Q_{ij}}$, $H_{R_{ij}}$, and $H_{t_{ij}}$are defined as follows:

$$\begin{cases} H_{F_{ij}} = J_{ij} \cdot {}_{I}^{L}R_{j|k-1} \cdot {}_{G}^{I}R_{j|k-1} \\ H_{Q_{ij}} = \mathrm{H}_{F_{ij}} \cdot \left[\left\lfloor {}^{G}P_{F_i} - {}^{G}P_{I\ j|k-1} \right\rfloor \quad -\mathrm{I}_3\right] \\ H_{R_{ij}} = J_{ij} \cdot \left[{}_{I}^{L}R_{j|k-1} \cdot \left\lfloor {}_{G}^{I}R_{j|k-1} \left({}^{G}P_{F_i} - {}^{G}P_{I\ j|k-1}\right)\right\rfloor \quad \mathrm{I}_3\right] \\ H_{t_{ij}} = J_{ij} \cdot {}_{I}^{L}R_{j|k-1} \\ \left[\left\lfloor {}^{I}\omega \right\rfloor \cdot {}_{G}^{I}R_{j|k-1} \left({}^{G}P_{F_i} - {}^{G}P_{I\ j|k-1}\right) \quad {}^{G}V_I(t)\right] \end{cases} \tag{33}$$

where J_{ij} is the Jacobian matrix; I_3denotes the 3-D identity matrix; the ${}_{L}^{I}R_{j|k-1}$, ${}_{G}^{I}R_{j|k-1}$, ${}^{G}P_{F_i}$, and ${}^{G}P_{I\,j|k-1}$ have been defined in (28)–(32).

Equation (33) is a linearized constraint of all LiDAR attitudes. These constraints express full information provided by $\overset{\approx}{\mathrm{X}}{}_{L_j}^{F_i}$.

5) State Updating: The LiDAR-IMU state vector Y_k and the state covariance matrix $\hat{C}_k$, combining with r^0, H^0,n^0 are used to conduct the MSCKF updating

$$\begin{cases} \mathrm{Y}_{k|k} = \mathrm{Y}_{k|k-1} + \Psi_k r^0 \\ \hat{C}_{k|k} = \hat{C}_{k|k-1} - \Psi_k S_k \Psi_k^T \end{cases} \tag{34}$$

where $S_k = \mathrm{H}^0 \hat{C}_{k|k-1} (\mathrm{H}^0)^T + n^0$, $\Psi_k = \hat{C}_{k|k-1} (\mathrm{H}^0)^T S_k^{-1}$, $r^0 = [r_1^0, \ldots, r_k^0, \ldots, r_m^0]^T$, $\mathrm{H}^0 = [\mathrm{H}_1^0, \ldots, \mathrm{H}_\mathrm{k}^0, \ldots, \mathrm{H}_\mathrm{m}^0]^T$,$n^0 = [n_1^0, \ldots, n_k^0, \ldots, n_m^0]^T$.

C. MSCKF Integrated DLRNN

The mainly characteristic of the MSCKF integrated DLRNN method are that the time delay error model and its optimal predictor can be realized by the DLRNN to form the improved state equation of the MSCKF; and then, the MSCKF can recursively provide an optimal estimation of the random drift and noise for the DLRNN. The random drift can be compensated in the training process. The MSCKF integrated DLRNN method is illustrated in Fig. 6.

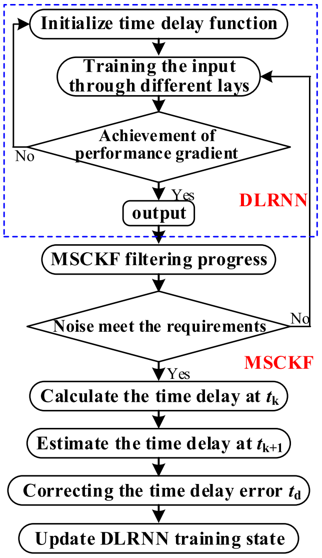
Initialize time delay function
Training the input through different lays
Achievement of performance gradient
No
Yes
output
DLRNN
MSCKF filtering progress
Noise meet the requirements
No
Yes
MSCKF
Calculate the time delay at t_k
Estimate the time delay at t_{k+1}
Correcting the time delay error t_d
Update DLRNN training state

Fig. 6. Algorithm of MSCKF integrated DLRNN.

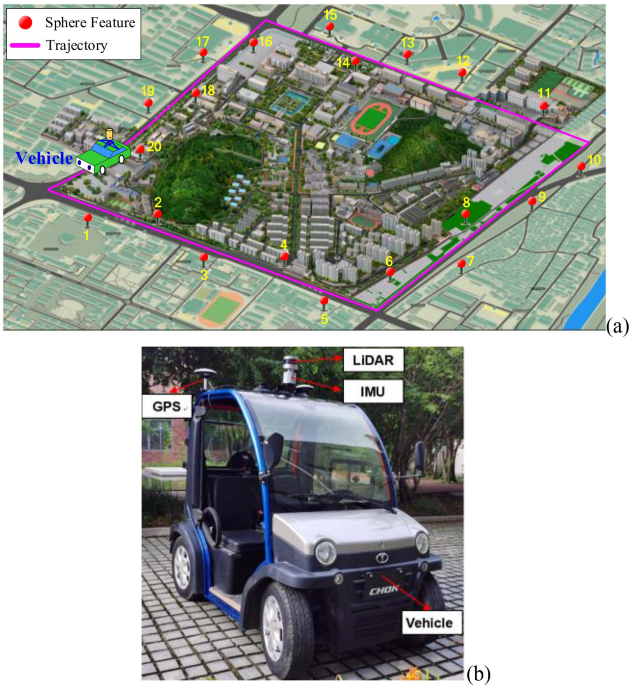

Fig. 7. Field test scenarios for time delay correction of LiDAR-IMU system. (a) Layouts of time delay correction experiments and sphere target feature layouts. (b) Mobile platform of a vehicle equipped with LiDAR, IMU, and GPS receiver.

IV. Experiments and Discussions

To evaluate the performance of the proposed method, a series of experiments were conducted in the university campus, where the campus shape was approximately square with a total area of 100 000 square meters. Fig. 7 shows the experimental layouts. The experiments were carried out using a Hesai Pandar64 LiDAR and a Spatial FOG1750 IMU rigidly mounded on the platform equipped in a vehicle. The specifications of the LiDAR

TABLE I
SPECIFICATIONS OF IMU AND LIDAR

Navigation Features	IMU	Performance Features	LiDAR
Horizontal Position Accuracy	0.5 m	Range	0 ~ 200 m
Vertical Position Accuracy	0.8 m	Distance accuracy	±2cm
Velocity Accuracy	0.007 m/s	Vertical FOV	40°
Roll & Pitch Accuracy	0.01°	Horizontal FOV	360°
Heading Accuracy	0.05°	Angle accuracy	0.167°
Sampling frequency	300 Hz	Sampling frequency	20 Hz

and IMU are given in Table I. In the experiments, aiming to verify the effectiveness of the proposed approach, the GPS, LiDAR, and IMU were all equipped in the vehicle. The GPS can provide the standard timestamp for the LiDAR-IMU system. The standard sphere targets observed by the LiDAR will be used as ground control targets to provide the actual position and orientation in GPS absence environments, their coordinates were established through total station systems with higher accuracy. They were uniformly and separately distributed in one side of the campus square along the vehicle trajectory at different ranges from around 6–20 m to provide sufficient observable accesses for the LiDAR-IMU. In addition, before conducting the tests, timestamps recorded by the LiDAR and IMU were aligned according to the GPS standard timestamp; by doing so, the timestamps of the LiDAR and IMU can be accurately aligned and the initial time delay error between LiDAR and IMU can be regarded as zero. Note that in a normal GPS environment, the timestamps of the LiDAR and IMU can be aligned based on the GPS timestamp; while in the GPS-absence scenarios, the timestamp alignment between the LiDAR and IMU can be initialized through the hardware setting.

In the experimental tests, three different groups of tests were conducted to evaluate the performance of the time delay error correction. The first group was to examine the online correction in the static situation of the LiDAR-IMU, the second one was tested in the dynamic situation, and the third one was to evaluate the positioning accuracy of the vehicle.

A. Time Delay Correction in Static Situation

In the static situation, the vehicle stopped in the GPS-friendly environment. The overall measuring time of the LiDAR and IMU was 1800 s. The IMU sampling rate was 300 Hz while the LiDAR sampling frequency was 20 Hz. To validate the effectiveness of the proposed method, the time delay correction performance was compared with the popular EKF, RBFNN, and RBFNN-EKF methods. The timestamp recorded by the GPS receiver was adopted as the true value in the tests. The comparison results are given in Fig. 8, where the horizontal axis denotes the measuring time, and the vertical axis represents the root-means-square error (RMSE) of the time delay.

In Fig. 8, the red-square line represents the time delay error between the LiDAR and IMU using the proposed method; the green-dot line is estimated by the RBFNN-EKF; the blue-triangle line is produced by the EKF; the black-del line is calculated by the RBFNN. As can be seen that the RMSE of the time delay is significantly reduced by the proposed method and the correction error is within 0.4 ms. The proposed method outperforms the RBFNN-EKF, EKF, and RBFNN methods with a RMSE improvement of 45.70%, 61.29%, and 64.20%, respectively.

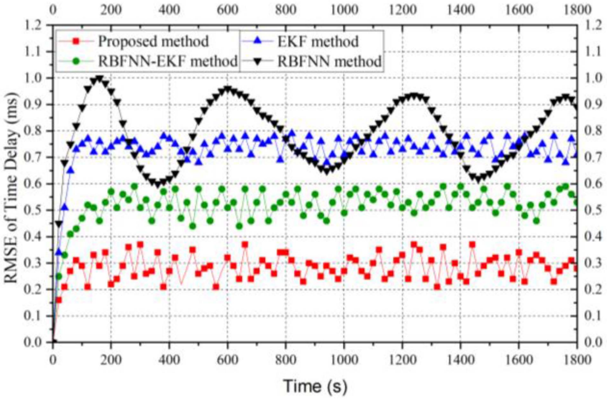

Fig. 8. RMSE of time delay in static situation.

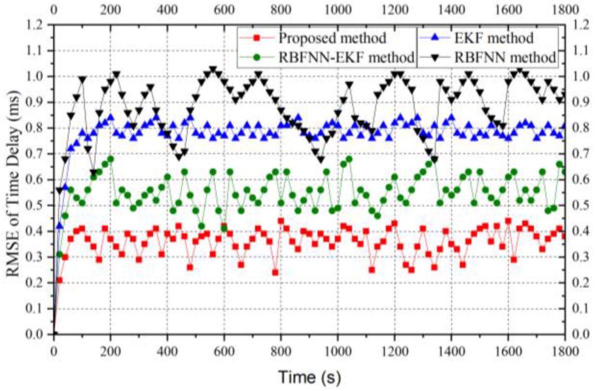

Fig. 9. Experimental trajectory of GPS absence.

B. Time Delay Correction in Dynamic Situation

In the dynamic situation, the vehicle moved repeatedly three times in both clockwise and counterclockwise around the runway of the campus square at variable speed with the best configuration parameters of the LiDAR and IMU. Twenty sphere targets with 140 mm in diameter were used as the LiDAR features. First of all, the timestamp of the LiDAR and IMU were accurately alignment based on the GPS standard timestamp; then the vehicle moved in variable speeds to verify the online correction performance of the time delay errors.

The IMU sampled at 300 Hz to acquire the vehicle position and pose, while the LiDAR sampled at 20 Hz to correct the vehicle position. The initial position of the vehicle was marked as the "start point," as shown in the Fig. 10, the measurements of the LiDAR and IMU were recorded for 1800 s, and the

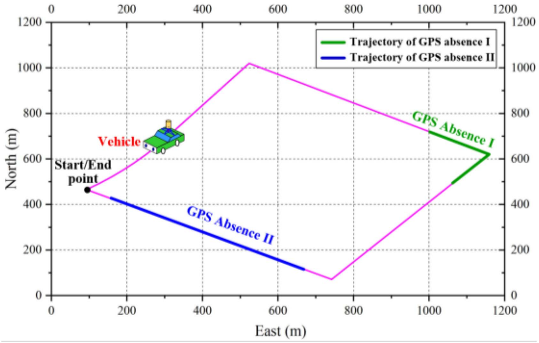

Fig. 10. RMSE of time delay in dynamic situation.

TABLE II
Computational Efficiency Comparison Among EKF, RBFNN, MSCKF, DLRNN, RBFNN-EKF, and Our Proposed Method for Tim e Delay Correction

Method	Mean Computing Time (ms)	Mean Training Accuracy RMSE (ms)
EKF	756.8	0.71
RBFNN	937.3	0.86
MSCKF	798.2	0.69
DLRNN	851.6	0.75
RBFNN-EKF	677.1	0.41
Proposed method	626.7	0.33

timestamps recorded by the GPS receiver were used as the true time-reference. Fig. 9 depicts the experimental results, which demonstrates that the proposed method is able to correct the time delay error in the dynamic situation and the estimated error is within 0.4 ms. Compared with the RBFNN-EKF, EKF, and RBFNN methods, the overall RMSE of the time delay correction using the proposed method improves 34.73%, 53.99%, and 59.25%, respectively.

C. Time Delay Error Correction Efficiency

To show the superiority of the proposed method, the time delay error correction efficiency was evaluated in the static situation and the obtained result was compared with the EKF, RBFNN, MSCKF, DLRNN, and RBFNN-EKF methods. During the comparison experiments, the computational time is determined by the training converge using the median over 10 tests.

The comparison results are shown in Table II, and it can be seen that the proposed method produces faster computational speed and higher training accuracy than that of the EKF, RBFNN, MSCKF, DLRNN, and RBFNN-EKF. The mean computing time using the proposed method reduces by 17.19%, 33.14%, 21.48%, 31.39%, and 8.44% against the other methods in Table II, and the mean training RMSE improves by 53.52%, 61.62%, 52.17%, 56%, and 19.51%, respectively.

D. Positioning Accuracy Estimation Experiments

To further evaluate the time delay correction performance of the proposed method in the GPS absence environments, the positioning accuracy estimation experiments were conducted. The time delay correction results using different methods are shown in Fig. 10. The GPS absence situations consisted of the turning movement and the straight movement, and the time of the GPA absence period I and period II are 300 s and 600 s, respectively.

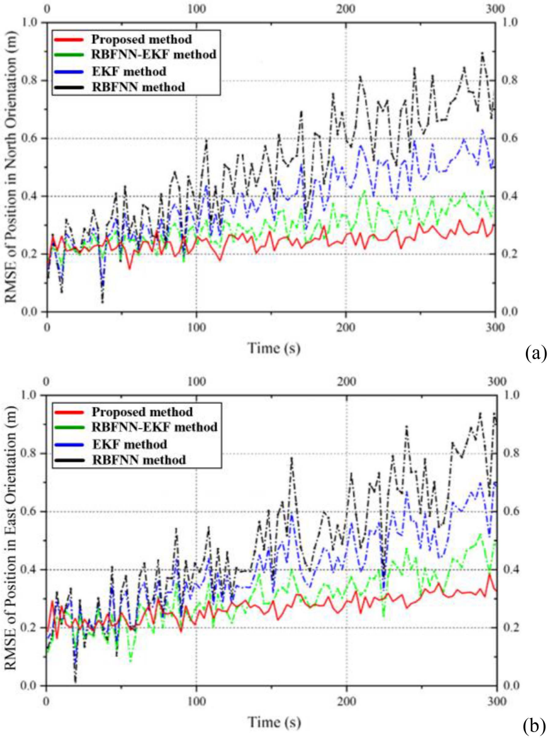

Fig. 11. RMSE of LiDAR-IMU position in GPS absence periods I. (a) North orientation. (b) East orientation.

1) Positioning Accuracy in GPS Absence Periods I: Fig. 11 depicts the positioning accuracy of the LiDAR-IMU in the North and East orientation at the GPS absence period I in the turning trajectory. It can be found that the proposed proposed method had a great advantage over other methods. During the first 50 s, the superiority of the proposed DLRNN-MSCKF method is not obvious to the EKF, RBFNN, and RBFNN-EKF methods; all the four methods provide good estimation on the positioning accuracy, which indicates that the position error can be well estimated by these methods during a very short GPS absence time in the turning movement. However, with the increase of the GPS absence time, the position errors estimated by the EKF, RBFNN, and RBFNN-EKF increase significantly; however, the proposed method is able to maintain good positioning accuracy. As a result, the proposed method in the turning movement can achieve higher positioning accuracy than the RBFNN-EKF, EKF, and RBFNN methods. The overall RMSE improvement of the positioning error of the proposed method over the other methods in the North orientation are, respectively, 25.39%, 46.59%, and 55.66%, and in the East orientation are, respectively, 25.76%, 42.35%, and 51.96%. Similarity, the overall standard deviation (Std) of the positioning accuracy improvement of the proposed method in the North orientation are, respectively, 39.77%, 49.52%, and 61.59%, and in the East orientation are, respectively, 36.26%, 42.57%, and 53.97%.

TABLE III
SPHERE TARGETS POSITION DETECTION COMPUTATIONAL BETWEEN EKF, RBFNN, RBFNN-EKF, AND OUR PROPOSED METHOD USING SPHERES NO.8–NO.12

ID	Proposed Method (cm)		RBFNN-EKF (cm)		EKF (cm)		RBFNN (cm)	
	2D	3D	2D	3D	2D	3D	2D	3D
S_8	1.08	1.68	1.30	1.87	1.66	2.51	2.13	2.62
S_9	1.53	1.64	1.71	2.01	2.17	2.36	2.66	3.35
S_{10}	1.43	1.77	1.57	1.86	2.31	2.65	2.71	3.56
S_{11}	1.33	1.59	1.52	1.79	2.23	2.59	2.62	3.29
S_{12}	1.45	1.79	1.69	1.97	1.86	2.50	2.16	2.75
Mean	1.36	1.69	1.55	1.90	2.04	2.52	2.45	3.11
Min	1.08	1.59	1.30	1.79	1.66	2.36	2.13	2.62
Max	1.53	1.79	1.71	2.01	2.31	2.65	2.71	3.56

TABLE IV
SPHERE TARGETS POSITION DETECTION COMPUTATIONAL BETWEEN EKF, RBFNN, RBFNN-EKF, AND OUR PROPOSED METHOD USING SPHERES NO.1–NO.5

ID	Proposed Method (cm)		RBFNN-EKF (cm)		EKF (cm)		RBFNN (cm)	
	2D	3D	2D	3D	2D	3D	2D	3D
S_1	1.35	1.53	1.61	1.83	2.04	2.29	2.46	2.98
S_2	0.92	1.28	1.46	1.90	1.94	2.41	2.43	2.93
S_3	0.96	1.01	1.33	1.41	1.61	1.83	2.23	2.81
S_4	1.32	1.58	2.01	2.61	2.29	2.97	2.61	3.23
S_5	1.48	1.96	1.64	2.13	2.19	2.87	2.67	3.38
Mean	1.20	1.47	1.61	1.97	2.01	2.47	2.48	3.06
Min	0.92	1.01	1.33	1.41	1.61	1.83	2.23	2.81
Max	1.48	1.96	2.01	2.61	2.29	2.97	2.67	3.38

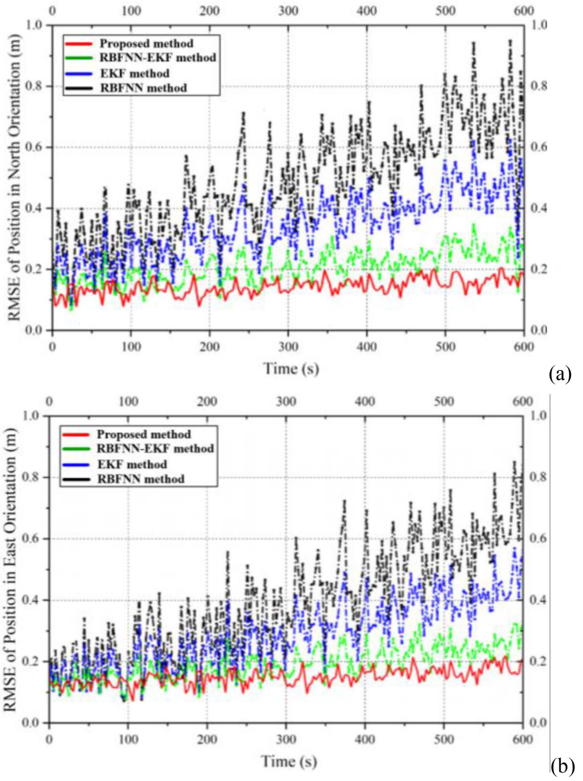

Fig. 12. RMSE of LiDAR-IMU position in GPS absence periods II. (a) North orientation. (b) East orientation.

In order to better understand the performance of time delay correction, the sphere targets No. 8–No. 12 in Fig. 7(a) were used in the position detection experiments during the GPS absence periods I.

The experimental results are given in Table III. It can be seen that the proposed method generates higher position detection accuracy for the sphere targets when compared with the RBFNN, EKF, and RBFNN-EKF, and the overall positioning accuracy improvements in 2-D and 3-D orientations are 24.02%, 16.73, 12.26% and 24.61%, 18.97%, 11.05%, respectively.

2) Positioning Accuracy in GPS Absence Periods II: As shown in Fig. 10, the trajectory of the LiDAR-IMU was run in the straight movement during the GPS absence periods II. In this experiment, we use the well-known Gaussian process regression method [18] to obtain the trajectory of LiDAR-IMU. Fig. 12 depicts the positioning accuracy in the period II, where the proposed method produces a great advance than the other methods during the straight movement tests. The DLRNN-MSCKF method can obtain better positioning accuracy during straight movement mode compared to the RBFNN-EKF, EKF, and RBFNN methods. The overall position error of the RMSE improvement of the proposed method over the others in the North orientation are, respectively, 25.00%, 48.15%, and 55.78%, and in the East orientation are, respectively, 25.86%,48.19%, and 52.75%. The overall Std of the positioning improvement of the proposed method over the others in the North orientation are, respectively, 40.25%, 50.53%, and 61.98%, and in the East orientation are, respectively, 37.04%, 46.87%, and 56.41%.

Similarity to the GPS absence periods I, the sphere targets No.1–No.5 were used to verify the performance of the proposed method in the GPS absence periods II. The experimental results in Table IV demonstrate that the proposed method achieves higher position detection accuracy of the sphere targets when compared with the RBFNN, EKF, and RBFNN-EKF methods; the overall positioning accuracy improvements in 2-D and 3-D orientations are 25.46%, 19.91, 18.92% and 25.38%, 20.24%, 19.28%, respectively.

V. CONCLUSION

In order to improve the online correction accuracy of the time delay error between the LiDAR and IMU sensors, a new DLRNN-MSCKF method is proposed. This new method possesses the advantages of both the DLRNN and MSCKF, which not only improves the training accuracy of the DLRNN, but also enhances the time delay correction performance of the MSCKF. Furthermore, the proposed method can accurately estimate the time delay error in the static and dynamic operation conditions of the LiDAR-IMU in a real-time manner and can be easily employed in different LiDAR-IMU frameworks. Experimental results demonstrate that the proposed method outperforms the RBFNN, EKF, and RBFNN-EKF methods in term of the estimation accuracy of the time delay error and produces better vehicle positioning accuracy in the turning and straight movements in the GPS absence environments.

Although the time delay correction accuracy for the LiDAR-IMU has been greatly improved using the proposed method, there are still some issues that should be addressed in the future plan; we outline some of them as follows: 1) investigation of the effect of multiple noise parameters in the training and estimating process of the DLRNN-MSCKF, which can further improve the time delay correction accuracy; 2) examination the performance of the proposed method to multitype sensors with varying time delay in different conditions (such as high speed, strong vibration, and strong disturbance environment).

Appendix

The matrixes of and in (7) can be specified as

$$A = \begin{bmatrix} -\lfloor \hat{\omega}_\times \rfloor & -I_3 & 0_{3\times3} & 0_{3\times3} & 0_{3\times3} & 0_{3\times7} \\ 0_{3\times3} & 0_{3\times3} & 0_{3\times3} & 0_{3\times3} & 0_{3\times3} & 0_{3\times7} \\ -{}^I_G R^T \lfloor \hat{a}_\times \rfloor & 0_{3\times3} & 0_{3\times3} & -{}^I_G R^T & 0_{3\times3} & 0_{3\times7} \\ 0_{3\times3} & 0_{3\times3} & I_3 & 0_{3\times3} & 0_{3\times3} & 0_{3\times7} \\ 0_{3\times3} & 0_{3\times3} & 0_{3\times3} & 0_{3\times3} & 0_{3\times3} & 0_{3\times7} \\ 0_{3\times3} & 0_{3\times3} & 0_{3\times3} & 0_{3\times3} & 0_{3\times3} & 0_{3\times7} \\ 0_{3\times3} & 0_{3\times3} & 0_{3\times3} & 0_{3\times3} & 0_{3\times3} & 0_{3\times7} \\ 0_{1\times3} & 0_{1\times3} & 0_{1\times3} & 0_{1\times3} & 0_{1\times3} & 0_{1\times7} \end{bmatrix} \tag{a1}$$

$$B = \begin{bmatrix} -I_3 & 0_{3\times3} & 0_{3\times3} & 0_{3\times3} \\ 0_{3\times3} & I_3 & 0_{3\times3} & 0_{3\times3} \\ 0_{3\times3} & 0_{3\times3} & -{}^I_G R^T & 0_{3\times3} \\ 0_{3\times3} & 0_{3\times3} & 0_{3\times3} & I_3 \\ 0_{3\times3} & 0_{3\times3} & 0_{3\times3} & 0_{3\times3} \\ 0_{3\times3} & 0_{3\times3} & 0_{3\times3} & 0_{3\times3} \\ 0_{3\times3} & 0_{3\times3} & 0_{3\times3} & 0_{3\times3} \\ 0_{1\times3} & 0_{1\times3} & 0_{1\times3} & 0_{1\times3} \end{bmatrix} \tag{a2}$$

where $\hat{\omega} = \omega_m - b_g$, $\hat{a} = a_m - b_a$.

References

[1] F. Aghili and C. Su, “Robust relative navigation by integration of ICP and adaptive Kalman filter using laser scanner and IMU,” *IEEE/ASME Trans. Mechatronics*, vol. 21, no. 4, pp. 2015–2026, Aug. 2016.

[2] W. Liu, Z. Li, R. Malekian, M. A. Sotelo, Z. Ma, and W. Li, “A novel multi-feature based on-site calibration method for LiDAR-IMU systems,” *IEEE Trans. Ind. Electron.*, vol. 67, no. 11, pp. 9851–9861, Nov. 2020.

[3] H. Shen and Y. Pan, “Improving tracking performance of nonlinear uncertain bilateral teleoperation systems with time-varying delays and disturbances,” *IEEE/ASME Trans. Mechatronics*, vol. 25, no. 3, pp. 1171–1181, Jun. 2020.

[4] Z. Lu, P. Huang, and Z. Liu, “Predictive approach for sensorless bimanual teleoperation under random time delays with adaptive fuzzy control,” *IEEE Trans. Ind. Electron.*, vol. 65, no. 3, pp. 2439–2448, Mar. 2018.

[5] G. Stienne, S. Reboul, M. Azmani, J. B. Choquel, and M. Benjelloun, “A multi-temporal multi-sensor circular fusion filter,” *Inform. Fusion.*, vol. 18, no. 1, pp. 86–100, Jul. 2014.

[6] S. Li, Y. Cheng, D. Brown, R. Tharmarasa, G. Zhou, and T. Kirubarajan, “Comprehensive time-offset estimation for multisensor target tracking,” *IEEE Trans. Aerosp. Electron. Syst.*, vol. 56, no. 3, pp. 2351–2373, Jun. 2020.

[7] S. Lee, M. Jeon, and V. Shin, “Distributed estimation fusion with application to a multisensory vehicle suspension system with time delays,” *IEEE Trans. Ind. Electron.*, vol. 59, no. 11, pp. 4475–4482, Nov. 2012.

[8] W. I. Liu and Y. Li, “Error modeling and extrinsic-intrinsic calibration for LiDAR-IMU system based on cone-cylinder features,” *Robot. Auton. Syst.*, vol. 114, pp. 124–133, Apr. 2019.

[9] A. K. Patil, A. Balasubramanyam, J. Y. Ryu, K. B. N. Pavan, and Y. H. Chai, “Fusion of multiple lidars and inertial sensors for the real-time pose tracking of human motion,” *Sensors*, vol. 20, no. 18, Sep. 2020, Art. no. 5342.

[10] H. Du, W. Wang, C. Xu, R. Xiao, and C. Sun, “Real-time onboard 3D State estimation of an unmanned aerial vehicle in multi-environments using multi-sensor data fusion,” *Sensors*, vol. 20, no. 3, Feb. 2020, Art. no. 919.

[11] H. Lee, S. Kang, and S. Han, “Real-time optimal State estimation scheme with delayed and periodic measurements,” *IEEE Trans. Ind. Electron.*, vol. 65, no. 7, pp. 5970–5978, Jul. 2018.

[12] J. Wu, M. Liu, C. Zhang, and Z. Zhou, “Correspondence matching and time delay estimation for hand-eye calibration,” *IEEE Trans. Instrum. Meas.*, vol. 69, no. 10, pp. 8304–8313, Oct. 2020.

[13] J. Kelly, N. Roy, and G. S. Sukhatme, “Determining the time delay between inertial and visual sensor measurements,” *IEEE Trans. Robot.*, vol. 30, no. 6, pp. 1514–1523, Dec. 2014.

[14] Y. Liu and Z. Meng, “Online temporal calibration based on modified projection model for visual-inertial odometry,” *IEEE Trans. Instrum. Meas.*, vol. 69, no. 7, pp. 5197–5207, Jul. 2020.

[15] X. Niu, T. Yu, J. Tang, and L. Chang, “An online solution of LiDAR scan matching aided inertial navigation system for indoor mobile mapping,” *Mobile Inf. Syst.*, vol. 2017, pp. 1–11, Jul. 2017.

[16] W. Liu, “LIDAR-IMU time delay calibration based on iterative closest point and iterated sigma point Kalman filter,” *Sensors*, vol. 17, no. 3, 2017, Art. no. 539.

[17] W. Li and H. Leung, “Simultaneous registration and fusion of multiple dissimilar sensors for cooperative driving,” *IEEE Trans. Intell. Transp. Syst.*, vol. 5, no. 2, pp. 84–98, Jun. 2004.

[18] G. A. Kumar, A. K. Patil, R. Patil, S. S. Park, and Y. H. Chai, “A LiDAR and IMU integrated indoor navigation system for UAVs and its application in real-time pipeline classification,” *Sensors*, vol. 17, no. 6, Jun. 2017, Art. no. 1268.

[19] Z. Taylor and J. Nieto, “Motion-based calibration of multimodal sensor extrinsic and timing offset estimation,” *IEEE Trans. Robot.*, vol. 32, no. 5, pp. 1215–1229, Oct. 2016.

[20] M. Zhang, X. Yu, P. Ding, L. Ou, and W. Zhang, “Distributed adaptive three-dimension formation control based on improved RBF neural network for non-linear multi-agent time-delay systems,” *IET Control Theory A*, vol. 13, no. 17, pp. 2758–2765, Nov. 2019.

[21] H. Ma, Z. Wang, D. Wang, D. Liu, P. Yan, and Q. Wei, “Neural-network-based distributed adaptive robust control for a class of nonlinear multiagent systems with time delays and external noises,” *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 46, no. 6, pp. 750–758, Jun. 2016.

[22] Y. Song, J. Hu, D. Chen, Y. Liu, F. E. Alsaadi, and G. Sun, “A resilience approach to state estimation for discrete neural networks subject to multiple missing measurements and mixed time-delays,” *Neurocomputing*, vol. 272, pp. 74–83, Jul. 2018.

[23] Q. Wang and J. Li, “Target tracking for a correlated multi-sensor system with constant time delay,” *Proc. Inst. Mech. Engineers, I, J. Syst. Control Eng.*, vol. 231, no. 9, pp. 753–761, Oct. 2017.

[24] R. Niu, P. K. Varshney, K. Mehrotra, and C. Mohan, “Temporally staggered sensors in multi-sensor target tracking systems,” *IEEE Trans. Aerosp. Electron. Syst.*, vol. 41, no. 3, pp. 794–808, Jul. 2005.

[25] D. Li, J. Zhou, and Y. Liu, “Recurrent-neural-network-based unscented Kalman filter for estimating and compensating the random drift of MEMS gyroscopes in real time,” *Mech. Syst. Signal Process.*, vol. 147, Jul. 2021, Art. no. 107057.

[26] Y. Chu, J. Fei, and S. Hou, “Adaptive global sliding-mode control for dynamic systems using double hidden layer recurrent neural network structure,” *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 31, no. 4, pp. 1297–1309, Apr. 2020.

[27] C. Shen, Y. Zhang, J. Tang, H. Cao, and J. Liu, “Dual-optimization for a MEMS-INS/GPS system during GPS outages based on the cubature Kalman filter and neural networks,” *Mech. Syst. Signal Process.*, vol. 133, Jul. 2019, Art. no. 106222.

[28] J. Fei and Y. Chen, “Dynamic terminal sliding-mode control for single phase active power filter using new feedback recurrent neural network,” *IEEE Trans. Power Electron.*, vol. 35, no. 9, pp. 9906–9924, Sep. 2020.

[29] Z. Zhang, Q. Wang, and Z. Zhang, “Harmonic vector error analysis based on lagrange interpolation,” *IEEE Access*, vol. 9, pp. 57464–57474, 2021.

[30] G. Wang, D. Yu, and P. Zhou, “Neural network interpolation operators optimized by Lagrange polynomial,” *Neural Netw.*, vol. 153, pp. 179–191, 2022.

[31] W. Bo, M. Bi, S. Xiao, J. Fang, T. Huang, and Y. Zhang, “Lagrange interpolation based extended Kalman filter for phase noise suppression in CO-OFDM system,” *Opt. Commun.*, vol. 435, pp. 221–226, Mar. 2019.

Wanli Liu received the Ph.D. degree in instrumentation engineering from Tianjin University, Tianjin, China, in 2016.

He is currently a Research Fellow of the Jiangsu Province and Education Ministry Co-Sponsored Collaborative Innovation Center of Intelligent Mining Equipment, China University of Mining and Technology, Beijing, China. He is an Editor of *International Journal of Mining Science and Technology* (JCR Q1, IF = 11.8). His current research interests include information fusion, LiDAR scanning, positioning and navigation, unmanned vehicle, and intelligent robotic.

Zhixiong Li (Senior Member, IEEE) received the Ph.D. degree in transportation engineering from the Wuhan University of Technology, Wuhan, China, in 2013.

He is currently an Associate Professor with the Faculty of Mechanical Engineering, Opole University of Technology, Opole, Poland. His research interests include intelligent vehicles and control, loop closure detection, and mechanical system modeling and control.

Dr. Li is an Associate Editor of IEEE Transactions on Intelligent Transportation Systems and a Column Editor of *IEEE Intelligent Transportation Systems Magazine*. He is a recipient of Fellow of the Institute of Physics (U.K.).

Weihua Li received the Ph.D. degree in mechanical engineering from Nanyang Technological University, Singapore, in 2001. He is a Senior Professor and Director of the Advanced Manufacturing Research Strength with the University of Wollongong, Wollongong, NSW, Australia. He has authored and coauthored more than 350 journal and conference papers with an h-index of 47 (Google Scholar). His research interests include smart materials and structures, dynamics and vibration control, microfluidics, Lab on a Chip, and vehicle technology.

Prof. Li is an editor or editorial board member for eleven international journals, including IEEE/ASME Transactions on Mechatronics*, Smart Materials and Structures, Lab on a Chip,* and *Journal of Intelligent Material Systems and Structures*. He is a recipient of Fellow of the Engineers Australia, the Fellow of Institute of Physics (U.K.), Australian Endeavour Research Fellowship, and JSPS Invitation Fellowship.

Paolo Gardoni received the Ph.D. degree in civil engineering from the University of California, Berkeley, Berkeley, CA, USA, in 2002.

He is the Alfredo H. Ang Family Professor and an Excellence Faculty Scholar with the Department of Civil and Environmental Engineering, University of Illinois at Urbana-Champaign, Champaign, IL, USA. He has authored and coauthored more than 170 refereed journal papers. His research interests include modeling of natural hazards.

Dr. Gardoni is an Editor-in-Chief of *Reliability Engineering and System Safety*. He is a Fellow of the Office of Risk Management and Insurance Research with the Gies College of Business, University of Illinois at Urbana-Champaign.

Haiping Du (Senior Member, IEEE) received the Ph.D. degree in mechanical design and theory from Shanghai Jiao Tong University, Shanghai, China, in 2002.

He was a Postdoctoral Research Associate with the Imperial College London, from 2003 to 2005, and with the University of Hong Kong, from 2002 to 2003. He was a Research Fellow with the University of Technology, Sydney, from 2005 to 2009. He is currently a Professor with the School of Electrical, Computer, and Telecommunications Engineering, University of Wollongong, Wollongong, NSW, Australia. His current research interests include robotics and automation, vehicle dynamics and control systems, and electric vehicles.

Dr. Du is an Associate Editor for IEEE Transactions on Industrial Electronics, a Subject Editor for the *Journal of Franklin Institute*, and an Associate Editor for *IEEE Control Systems Society Conference Editorial Board*.

Miguel Angel Sotelo (Fellow, IEEE) received the Ph.D. degree in electrical engineering from the University of Alcalá (UAH), Madrid, Spain, in 2001.

He is currently a Full Professor with the Department of Computer Engineering, UAH. He was a Project Evaluator, Rapporteur, and Reviewer for the European Commission in the field of ICT for Intelligent Vehicles and Cooperative Systems in FP6 and FP7. His research interests include autonomous vehicles and robotics.

Dr. Sotelo is member of the IEEE ITSS Board of Governors and Executive Committee. He was an Editor-in-Chief of the *IEEE Intelligent Transportation Systems Magazine*, Editor-in-Chief of the *ITSS Newsletter,* and Associate Editor of IEEE Transactions on Intelligent Transportation Systems. At present, he is past-President of the *IEEE Intelligent Transportation Systems Society*.