3DIO: Low-Drift 3-D Deep-Inertial Odometry for Indoor Localization Using an IMU

Shiyu Bai, *Member, IEEE*, Weisong Wen, *Member, IEEE*, and Chuang Shi

***Abstract*—The use of mobile devices for indoor localization has proven to be a convenient solution for pedestrians in Internet of Things (IoTs) applications. Radiofrequency (RF) signals, including Wi-Fi, Bluetooth, and others, are among the most commonly used sources. However, their availability cannot be guaranteed in all scenarios. Although pedestrian dead reckoning (PDR) using an inertial measurement unit (IMU) provides a self-contained positioning solution, it is susceptible to error accumulation due to heading uncertainties and varying motions. This article presents a low-drift 3-D deep-inertial odometry (DIO) method for indoor pedestrian localization using an IMU. The proposed approach employs a neural network to regress speeds within the human body frame, ensuring that the speeds are unaffected by absolute heading. These regressed speeds are integrated with inertial navigation to determine position. To enhance accuracy, the method incorporates an invariant extended Kalman filter (InEKF)-based integration for state estimation. Additionally, a learned height is included in the filter to improve 3-D position estimation. The performance of the proposed method is validated through real-world tests in various environments. Results demonstrate that the proposed method outperforms traditional PDR, robust neural inertial navigation (RONIN), and EKF-based techniques. Furthermore, this article examines the method from multiple perspectives, highlighting its strengths in addressing heading drift and varying motions, as well as the impact of height constraints and behavior-based position corrections. The video (YouTube) or (BiliBili) is shared to showcase our work.**

***Index Terms*—3-D space, deep learning, indoor localization, inertial measurement unit (IMU), inertial odometry, invariant extended Kalman filter (InEKF).**

I. Introduction

INDOOR localization has become a fundamental feature in Internet of Things (IoTs) applications, such as healthcare [1], virtual/augmented reality [2], and emergency response [3]. Currently, radiofrequency (RF) signals serve as the primary source for indoor localization. Common RF signals include Wi-Fi, Bluetooth low energy (BLE), ultrawideband (UWB), and 5G [4], [5], [6]. Corresponding indoor localization methods have been extensively studied, with their performance steadily improving. However, the deployment of RF infrastructure is required for these methods, and it may be limited by cost constraints. Although Wi-Fi and BLE are widely available in indoor environments and do not require additional device installation, periodic calibration is often necessary. Furthermore, multipath effects and non Line-of-Sight (NLoS) conditions in indoor environments present significant challenges, restricting the performance of RF-based indoor localization.

The navigation method based on an inertial measurement unit (IMU) provides continuous and self-contained positioning, compensating for localization gaps when RF signals are unavailable. Inertial navigation is a widely used approach; however, it suffers from significant drift, particularly in indoor applications that rely on low-cost micro-electromechanical system (MEMS) IMUs [7]. pedestrian dead reckoning (PDR) offers superior localization performance compared to traditional inertial navigation [8], as it limits error accumulation through a single integration operation. However, classical PDR faces two primary challenges. First, it is prone to heading uncertainties. While gyroscopes are commonly used to determine heading, the presence of unknown biases can negatively impact heading accuracy. The addition of a magnetometer has become a common solution, but magnetic disturbances often impair the reliability of heading estimation. Second, PDR lacks adaptability to varying motions. In PDR, step length is typically calculated using predefined models [9], which are designed to handle only a single motion pattern. However, pedestrians may exhibit diverse movements, such as walking on flat surfaces or ascending and descending stairs. In such cases, traditional step-length estimation can lead to substantial errors. Recently, deep learning-based inertial odometry methods have been developed to overcome the limitations of PDR. These methods are implemented in various forms, including pure end-to-end models and models integrated with inertial navigation. However, they still experience considerable drift, particularly in 3-D indoor environments.

In conclusion, the limitations of existing methods can be summarized as follows: on the one hand, current end-to-end approaches tend to transform IMU data directly into the world frame. While roll and pitch can be obtained relatively easily, achieving reliable heading estimation remains challenging due to unknown biases. When heading errors become significant,

Received 16 September 2024; revised 10 November 2024; accepted 19 December 2024. Date of publication 23 December 2024; date of current version 25 April 2025. This work was supported in part by the Guangdong Basic and Applied Basic Research Foundation under Grant 2021A1515110771; in part by the University Grants Committee of Hong Kong under the Scheme Research Impact Fund under Grant R5009-21; and in part by the Faculty of Engineering, The Hong Kong Polytechnic University under the project "Perception-based GNSS PPP-RTK/LVINS integrated navigation system for unmanned autonomous systems operating in urban canyons." *(Corresponding author: Weisong Wen.)*

Shiyu Bai and Weisong Wen are with the Department of Aeronautical and Aviation Engineering, The Hong Kong Polytechnic University, Hong Kong, China (e-mail: shiyu.bai@polyu.edu.hk; welson.wen@polyu.edu.hk).

Chuang Shi is with the School of Electronic and Information Engineering, Beihang University, Beijing 100083, China (e-mail: shichuang@buaa.edu.cn).

Digital Object Identifier 10.1109/JIOT.2024.3521404

the performance of these end-to-end approaches deteriorates considerably. On the other hand, although the impact of heading errors can be mitigated by using a filter to estimate the bias, the existing methods typically rely on the extended Kalman filter (EKF). However, EKF-based approaches are prone to nonlinear errors, resulting in substantial error accumulation in the odometry over time.

To address these challenges, this article proposes a novel 3-D indoor localization method using only the IMU embedded in a smartphone. The main contributions of this article are summarized as follows.

1) Unlike traditional methods, this article employs deep learning to estimate both the forward and vertical speeds of the pedestrian from IMU data. This approach ensures that the regression is independent of the absolute heading, thereby enhancing the accuracy of the learned motion estimation.
2) This article presents a 3-D deep-inertial odometry (3DIO) method based on the invariant extended Kalman filter (InEKF). The learned speeds are integrated with strap-down inertial navigation to perform state estimation. Additionally, a learned height constraint is introduced to reduce vertical errors. The proposed method effectively mitigates errors caused by heading drift and nonlinearity in 3-D scenarios.
3) Real-world tests are conducted across various indoor scenarios, demonstrating the proposed method's advantages in addressing heading drift, vertical errors, and varying motions. Additionally, the extension of the 3DIO, supported by a behavior map, is explored.

The remainder of this article is organized as follows. Section II reviews related work. Section III provides an overview of the proposed method, while Sections IV and V present the method in detail. Section VI reports the experimental results and discussions. Finally, Section VII concludes this article.

II. Related Work

Currently, RF-based techniques are the mainstream approach for indoor localization, with signal coverage serving as a prerequisite for their functionality. Among RF signals, Wi-Fi and BLE are more prevalent in indoor environments, eliminating the need for additional device installations and providing ubiquitous localization capabilities for pedestrians. Fingerprinting is a common technique for Wi-Fi and BLE-based indoor localization, involving both training and positioning processes [10]. However, the need for periodic recalibration increases labor costs and workload. Researchers have also developed indoor localization methods utilizing smartphone built-in cameras [11], [12], [13]. However, these approaches tend to consume more power and are highly susceptible to changes in lighting conditions.

PDR is a popular method for indoor localization. It operates solely on an IMU, does not depend on any infrastructure, and is less affected by external environments. This greatly enhances the convenience and reduces the cost of usage. PDR includes the step detection, step length estimation, and heading estimation [14]. Step detection refers to the process of counting the number of steps. The methods for detection include threshold, peak detection, correlation, spectral analysis, and machine learning [15]. Step length estimation involves computing a pedestrian's displacement using the results from the detection process. Díez et al. [16] conducted a review of step estimation methods that utilize inertial sensors. Model-based approaches, which include constant, linear, and empirical models, are typically employed to obtain the step length. However, these traditional methods are typically designed to handle single motion patterns. When the pedestrian's walking mode changes, the accuracy of the estimated step length tends to deteriorate. To solve this issue, adaptive step estimation methods have been proposed. Martinelli et al. [17] developed a weighted context-based step length estimation method for PDR. Six contexts are considered: 1) stationary; 2) walking; 3) walking sideways; 4) climbing stairs; 5) descending stairs; and 6) running. The step length is weighted according to the probabilities of these contexts. Yao et al. [18] proposed a walking pattern-aware step length estimation method. The pedestrian's walking pattern is recognized using a random forest with classification proofreading. Based on this recognition, the corresponding step length model is then selected. Heading estimation is another critical factor that affects the performance of PDR. Normally, external aiding information is integrated to mitigate the effect of unknown bias in gyroscopes. Yang et al. [19] presented a robust method for estimating heading in smartphone-based indoor localization. An EKF is designed to adaptively fuse IMU data, corrected magnetic headings, and building map headings. To achieve a comprehensive correction, Choi and Choi [20] introduced an online calibration for PDR parameters. This method employs a Kalman filter (KF) to estimate the step length, initial position, and reference heading direction. While the inclusion of external sources can achieve more accurate estimations of step length and heading, their effectiveness is limited when the external sources are either highly erroneous or unavailable.

To overcome the limitations of traditional PDR, learning-based methods have been introduced. Chen et al. [21] introduced an inertial odometry neural network (IONet), which utilizes a bidirectional long short-term memory (LSTM) [22] to calculate changes in distance and heading based on a sequence of raw IMU data. The test results demonstrate that it surpasses the performance of step-based PDR and inertial navigation. Yan et al. [23] proposed a robust IMU double integration (RIDI). It employs support vector regression (SVR) to derive the velocity vector from historical linear accelerations and angular velocities. The regressed velocity is then utilized to rectify the error in the linear acceleration. This corrected linear acceleration is subsequently used to estimate the position through double integration. Herath et al. [24] presented a robust neural inertial navigation (RONIN), a method that regresses horizontal displacements and body heading. RONIN is implemented using three different architectures: 1) ResNet [25]; 2) LSTM; and 3) temporal convolutional network (TCN) [26]. In RONIN, IMU data needs to be first transformed into the global frame, which makes it easily affected by orientation accuracy. Wang et al. [27] proposed

a pose-invariant inertial odometry. A random orientation initialization is utilized during training to regress the velocity. This approach can alleviate the dependency on the orientation of the smartphone. Furthermore, enhancing the generalization capability is also a crucial subject. Cao et al. [28] proposed a rotation-equivariance supervised inertial odometry (RIO). The implementation of a self-supervision scheme can diminish the dependency on extensive amounts of labeled data for training. Furthermore, they proposed an adaptive test-time training (TTT) approach, which is based on uncertainty estimations, to enhance generalizability for unseen data. Apart from the pure end-to-end method, researchers also integrate learning with inertial navigation to accomplish state estimation. Liu et al. [29] developed a tight learned inertial odometry (TLIO), which employs a ResNet-based network to regress 3-D displacements and their corresponding uncertainties. Subsequently, a stochastic cloning EKF is used to estimate the pose, velocity, and sensor bias. This method has been shown to provide superior trajectory estimation compared to RONIN. To improve the applicability in mobile devices, Wang et al. [30] introduced a lightweight learned inertial odometer (LLIO). A lightweight multilayer perceptron (MLP)-based network has been designed for regression. LLIO demonstrates a level of accuracy like that of TLIO, but with a remarkable improvement in efficiency.

TLIO and its extensions, collectively known as DIO, effectively leverage both model-based and data-driven approaches for reliable state estimation. Additionally, these methods can estimate sensor bias, further enhancing localization performance. Consequently, DIO proves effective for localization in scenarios where external localization data is unavailable. In state estimation, optimization-based methods have recently gained popularity across various fields, including simultaneous localization and mapping (SLAM) [31], urban navigation [32], and indoor localization [33]. While optimization-based methods offer superior performance compared to traditional filtering-based approaches [34], they tend to consume more power, posing a significant challenge for mobile devices used in localization. As a result, filtering-based methods, primarily the EKF, remain the primary tool for sensor fusion. However, the classical EKF relies on the current state estimate for linearization. If the estimate deviates significantly from the true state, the linearization process introduces errors, potentially creating a positive feedback loop that can lead to filter divergence [35].

Recent research has demonstrated the benefits of using Lie Groups for state estimation. The InEKF, an extension of the EKF, enhances estimation accuracy by being independent of the current state estimate. Brossard et al. [36] introduced a robust inertial navigation system on wheels (RINS-Ws). They designed a detector based on a recurrent deep neural network to dynamically identify situations of interest for the vehicle, such as zero velocity or no lateral slip. The InEKF was employed to fuse inertial navigation data with the detected pseudo-measurements. Evaluations on a public dataset demonstrated that the proposed method outperforms pure inertial navigation, standalone wheeled odometry, and wheeled odometry using fiber-optic gyros (FOGs). Hartley et al. [37] proposed a contact-aided InEKF for state estimation in bipedal robots. Their approach combined contact-inertial dynamics with forward kinematic corrections to estimate the robot's state, showing that the InEKF outperforms the traditional quaternion-based EKF. Potokar et al. [38] applied the InEKF to underwater localization, integrating data from the IMU, doppler velocity log (DVL), and pressure sensors. A Monte Carlo simulation demonstrated that the InEKF outperforms the quaternion-based EKF in this context as well.

III. Methodology Overview

The overall structure of the proposed method is illustrated in Fig. 1. The method utilizes 6-axis IMU data from a smartphone, consisting of acceleration and angular velocity, as input. This IMU data is applied to strap-down inertial navigation to predict the current state. Simultaneously, the IMU data is aligned with the pedestrian's orientation using the estimated attitude and then fed into a neural network to regress the pedestrian's forward and vertical speeds. A nonholonomic constraint (NHC) is employed, assuming the pedestrian has no lateral speed. The forward and vertical speeds are integrated with the NHC to generate a 3-D velocity observation. At the same time, the vertical speed is used to calculate the height through dead reckoning, forming a height observation. The 3-D velocity and height observations are ultimately fused with strap-down inertial navigation within an InEKF to achieve state estimation. The estimated state includes the smartphone's attitude, velocity, position, gyroscope bias, and accelerometer bias. It is worth noting that this process can operate independently. The method can also be further enhanced by incorporating a behavior map, in which stair-taking behaviors are used for position correction, helping to limit the accumulation of errors.

To offer a clearer description of the proposed method, this article also introduces several frames. These frames are used to illustrate the spatial transformation relationships. In this article, the frames include the world frame, the smartphone frame, and the body frame. These are symbolized by w, s, and b, respectively, as depicted in Fig. 2. The world frame is used as the reference for navigation. In the smartphone frame, the x_s and y_s axes are aligned with the longitudinal and lateral axes, respectively. The z_s axis is positioned perpendicularly to the screen of the smartphone. The x_b and y_b axes of the body frame are directed toward the front and left of the human body, respectively. It is assumed that the plane formed by the x_b and y_b axes is parallel to the ground. The z_b axis is perpendicular to this plane. Regarding the symbols in this article, $\tilde{\mathbf{a}}$ denotes the variable from the sensor output and observation, while $\hat{\mathbf{d}}$ is the estimated variable. $\mathbf{d}$ is a general symbol.

IV. Learning-Based Forward and Vertical Speeds

In the proposed method, raw IMU data is used to regress the pedestrian's forward and vertical speeds. These regressed speeds are integrated with the NHC to serve as observations, helping to limit error drift in inertial navigation. This approach is independent of the absolute heading, minimizing its influence on the regression.

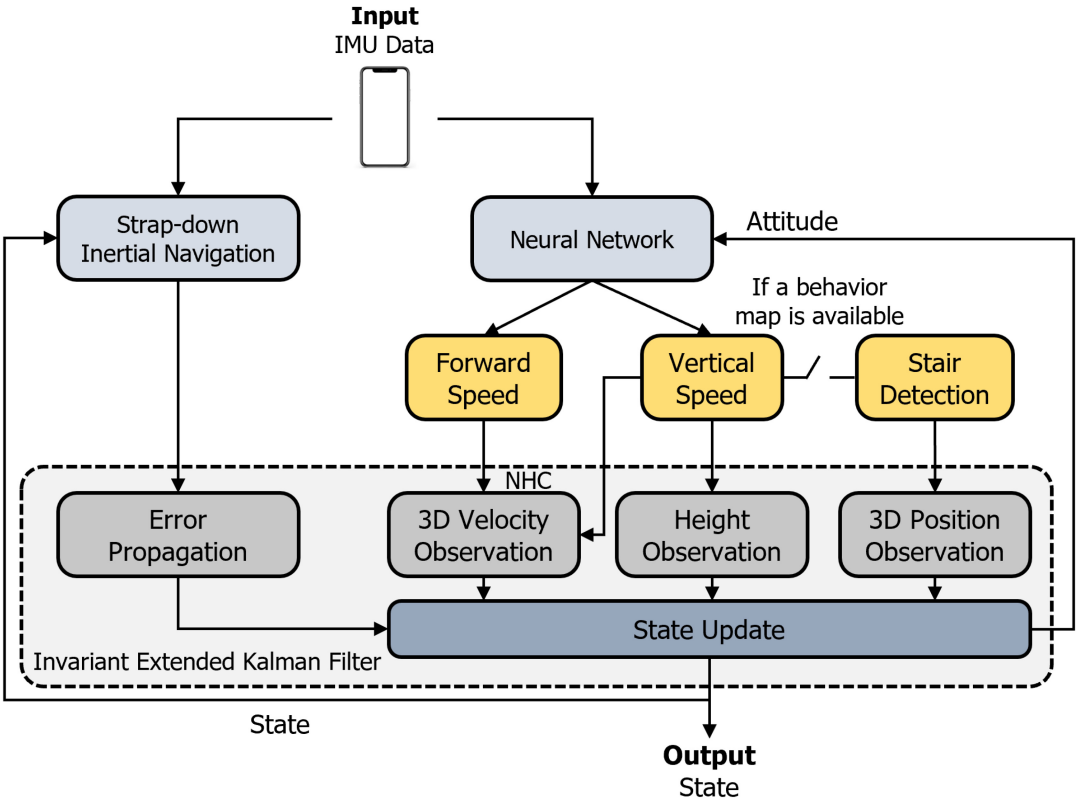

Fig. 1. Structure of the proposed method, in which the input is the 6-axis IMU measurements (acceleration and angular velocity) from a smartphone. The output includes the smartphone's attitude, velocity, position, gyroscope bias, and accelerometer bias.

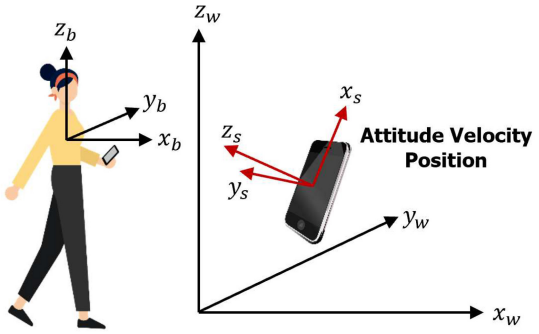

Fig. 2. Illustration of frames in this article.

For training, we collected our own dataset consisting of over 30 sequences, totaling 8 h, using two iPhones. One iPhone was used to collect gyroscope and accelerometer data at a frequency of 100 Hz. The other iPhone ran Apple ARKit, a visual-inertial odometry (VIO) function, to capture the human's trajectory in the world frame. The velocity in the world frame is determined through differential operations, and they are represented as v^{wx}, v^{wy}, and v^{wz}, respectively. v^{wz} is the vertical speed, denoted as v^{bv}. The forward speed can be calculated by determining the norm of v^{wx} and v^{wy}, which is expressed as

$$v^{bf} = \sqrt{(v^{wx})^2 + (v^{wy})^2}. \tag{1}$$

Raw IMU data is represented in the smartphone frame. To enhance regression performance, raw IMU data needs to be transformed from the smartphone frame to the body frame. This transformation aids in reducing the effects of the smartphone's varying attitudes, which can be expressed as

$$\begin{aligned} \hat{\mathbf{a}}_t^b &= \hat{\mathbf{R}}_s^b \tilde{\mathbf{a}}_t \\ \hat{\mathbf{w}}_t^b &= \hat{\mathbf{R}}_s^b \tilde{\mathbf{w}}_t \end{aligned} \tag{2}$$

where $\tilde{\mathbf{a}}_t$ and $\tilde{\mathbf{w}}_t$ are accelerometer and gyroscope data at the time instant t from the IMU. $\hat{\mathbf{R}}_s^b$ denotes the estimated rotation matrix from the smartphone frame to the body frame. $\hat{\mathbf{a}}_t^b$ and $\hat{\mathbf{w}}_t^b$ are the transformed accelerometer and gyroscope data in the body frame at the time instant t. In $\hat{\mathbf{R}}_s^b$, the angles along the x and y axes are obtained from the smartphone's roll and pitch, which are calculated in the filter. The angle along the z axis in $\hat{\mathbf{R}}_s^b$ is typically difficult to obtain as the smartphone can be held in various modes by the user. In this research, we only consider situations where the smartphone is relatively static with respect to the human body. Under these conditions, we can determine the angle along the z axis. In addition, the gravity vector is removed from $\hat{\mathbf{a}}_t^b$ to retain only the linear acceleration $\hat{\mathbf{l}}_t^b$.

This article uses a 1-D version of the ResNet-18 architecture, appending a fully connected (FC) layer at the end to estimate the forward and vertical speeds of the pedestrian. The network's input dimension is $n \times 6$. In this article, a sequence of data spanning one second is fed into the neural network, resulting in a value of 100 for n. The regression of forward and vertical speeds can be expressed as follows:

$$\mathbf{v}^b = \left(v^{bf}, v^{bv}\right) = f_{\text{ResNet}}\left(\hat{\mathbf{l}}_{1:n}^b,\ \hat{\mathbf{w}}_{1:n}^b\right) \tag{3}$$

where $\hat{\mathbf{l}}^b_{1:n}$ and $\hat{\mathbf{w}}^b_{1:n}$ are the sequences of linear acceleration and angular velocity, respectively.

The loss function is defined in the form of mean square error (MSE), as expressed below

$$\mathcal{L}_{\text{MSE}} = \frac{1}{N}\sum_{i=1}^{N}\left\|\hat{\mathbf{v}}^b_i - \mathbf{v}^b_i\right\|^2 \tag{4}$$

where N denotes the amount of data in the training dataset. During the model training, PyTorch is used to implement the model, and the training process is conducted using an NVIDIA GeForce RTX 4090. The Adam optimizer [39] is employed with an initial learning rate of 0.0001, no weight decay, and dropout layers with a probability of 0.5 for the FC layers. The models undergo training for a total of 5000 epochs.

V. InEKF for DIO

By utilizing deep learning, we can obtain the pedestrian's speeds expressed in the body frame. To compute the position, the absolute heading must be applied to project these velocities from the body frame to the world frame. However, since the heading is determined through the integration of gyroscope data, it is prone to drift due to unknown sensor bias. The velocity in the body frame can be used as an observation in a KF, enabling the online estimation of sensor bias and improving the accuracy of the heading estimation. Currently, the EKF is widely used for this type of fusion. However, the EKF often introduces larger errors due to nonlinearities, especially in odometry, where position is estimated through dead reckoning. Even small errors at a single time instant can accumulate, leading to significant positional errors over time. To address these challenges, this article proposes a deep-inertial odometry (DIO) method based on the InEKF. This section provides a detailed explanation of state prediction and measurement updates within the InEKF framework.

A. State Prediction

The state variable in this article is defined as

$$\mathbf{X}_k = \left(\mathbf{R}_k, \mathbf{v}_k, \mathbf{p}_k, \mathbf{b}^w_k, \mathbf{b}^a_k\right) \tag{5}$$

where $\mathbf{X}_k$ denotes the state vector at the time instant t_k, which includes the rotation, velocity, position, gyroscope bias, and accelerometer bias. The differential equation of the state can be expressed as

$$\begin{aligned}
\dot{\mathbf{R}}_t &= \mathbf{R}_t\left[\tilde{\mathbf{w}}_t - \mathbf{b}^w_t - \mathbf{n}^w_t\right]_\times \\
\dot{\mathbf{v}}_t &= \mathbf{R}_t\left(\tilde{\mathbf{a}}_t - \mathbf{b}^a_t - \mathbf{n}^a_t\right) + \mathbf{g} \\
\dot{\mathbf{p}}_t &= \mathbf{v}_t \\
\dot{\mathbf{b}}^w_t &= \mathbf{n}^{b_w}_t \\
\dot{\mathbf{b}}^a_t &= \mathbf{n}^{b_a}_t
\end{aligned} \tag{6}$$

where $\mathbf{n}^w_t$ and $\mathbf{n}^a_t$ represent the noise of the gyroscope and accelerometer, respectively. These noises are assumed to follow a Gaussian distribution, denoted as $\mathcal{N}(\mathbf{0}, (\boldsymbol{\sigma}^w_t)^2)$ and $\mathcal{N}(\mathbf{0}, (\boldsymbol{\sigma}^a_t)^2)$. $\mathbf{g}$ denotes the gravity vector in the world frame. The biases of the gyroscope and accelerometer are modeled as a random walk, with their derivatives represented as $\mathbf{n}^{b_w}_t$ and $\mathbf{n}^{b_a}_t$. These derivatives are also assumed to follow a Gaussian distribution, denoted as $\mathcal{N}(\mathbf{0}, (\boldsymbol{\sigma}^{b_w}_t)^2)$ and $\mathcal{N}(\mathbf{0}, (\boldsymbol{\sigma}^{b_a}_t)^2)$.

In the state variable, the rotation, velocity, and position can be organized to form the matrix Lie group. Consequently, the new state variable can be expressed as follows:

$$\begin{aligned}
\boldsymbol{\chi}_t &= \begin{pmatrix} \mathbf{R}_t & \mathbf{v}_t & \mathbf{p}_t \\ \mathbf{0}_{1\times 3} & 1 & 0 \\ \mathbf{0}_{1\times 3} & 0 & 1 \end{pmatrix} \\
\boldsymbol{\theta}_t &= \begin{pmatrix} \mathbf{b}^w_t \\ \mathbf{b}^a_t \end{pmatrix}.
\end{aligned} \tag{7}$$

The error of $\boldsymbol{\chi}_t$ in right and left invariant form can be expressed as

$$\begin{aligned}
\boldsymbol{\eta}^r_t &= \hat{\boldsymbol{\chi}}_t \boldsymbol{\chi}_t^{-1} \\
\boldsymbol{\eta}^l_t &= \boldsymbol{\chi}_t^{-1} \hat{\boldsymbol{\chi}}_t
\end{aligned} \tag{8}$$

where

$$\boldsymbol{\chi}_t^{-1} = \begin{pmatrix} \mathbf{R}_t^{\mathrm{T}} & -\mathbf{R}_t^{\mathrm{T}}\mathbf{v}_t & -\mathbf{R}_t^{\mathrm{T}}\mathbf{p}_t \\ \mathbf{0}_{1\times 3} & 1 & 0 \\ \mathbf{0}_{1\times 3} & 0 & 1 \end{pmatrix}. \tag{9}$$

According to (7) and (9), the $\boldsymbol{\eta}^r_t$ and $\boldsymbol{\eta}^l_t$ can be denoted as

$$\begin{aligned}
\boldsymbol{\eta}^r_t &= \begin{pmatrix} \hat{\mathbf{R}}_t\mathbf{R}_t^{\mathrm{T}} & \hat{\mathbf{v}}_t - \hat{\mathbf{R}}_t\mathbf{R}_t^{\mathrm{T}}\mathbf{v}_t & \hat{\mathbf{p}}_t - \hat{\mathbf{R}}_t\mathbf{R}_t^{\mathrm{T}}\mathbf{p}_t \\ \mathbf{0}_{1\times 3} & 1 & 0 \\ \mathbf{0}_{1\times 3} & 0 & 1 \end{pmatrix} \\
&= \begin{pmatrix} \boldsymbol{\eta}^r_{R_t} & \boldsymbol{\xi}^r_{v_t} & \boldsymbol{\xi}^r_{p_t} \\ \mathbf{0}_{1\times 3} & 1 & 0 \\ \mathbf{0}_{1\times 3} & 0 & 1 \end{pmatrix}
\end{aligned} \tag{10}$$

$$\begin{aligned}
\boldsymbol{\eta}^l_t &= \begin{pmatrix} \mathbf{R}_t^{\mathrm{T}}\hat{\mathbf{R}}_t & \mathbf{R}_t^{\mathrm{T}}\hat{\mathbf{v}}_t - \mathbf{R}_t^{\mathrm{T}}\mathbf{v}_t & \mathbf{R}_t^{\mathrm{T}}\hat{\mathbf{p}}_t - \mathbf{R}_t^{\mathrm{T}}\mathbf{p}_t \\ \mathbf{0}_{1\times 3} & 1 & 0 \\ \mathbf{0}_{1\times 3} & 0 & 1 \end{pmatrix} \\
&= \begin{pmatrix} \boldsymbol{\eta}^l_{R_t} & \boldsymbol{\xi}^l_{v_t} & \boldsymbol{\xi}^l_{p_t} \\ \mathbf{0}_{1\times 3} & 1 & 0 \\ \mathbf{0}_{1\times 3} & 0 & 1 \end{pmatrix}.
\end{aligned} \tag{11}$$

Let $\boldsymbol{\xi}^r_{R_t}$ and $\boldsymbol{\xi}^l_{R_t}$ be the Lie algebras of $\boldsymbol{\eta}^r_{R_t}$ and $\boldsymbol{\eta}^l_{R_t}$, respectively. Then, $\boldsymbol{\eta}^r_{R_t}$ and $\boldsymbol{\eta}^l_{R_t}$ can be expressed as

$$\begin{aligned}
\boldsymbol{\eta}^r_{R_t} &= \mathbf{I} + \left[\boldsymbol{\xi}^r_{R_t}\right]_\times \\
\boldsymbol{\eta}^l_{R_t} &= \mathbf{I} + \left[\boldsymbol{\xi}^l_{R_t}\right]_\times.
\end{aligned} \tag{12}$$

The error of IMU bias can be expressed as

$$\boldsymbol{\varsigma}_t = \begin{pmatrix} \hat{\mathbf{b}}^w_t - \mathbf{b}^w_t \\ \hat{\mathbf{b}}^a_t - \mathbf{b}^a_t \end{pmatrix} = \begin{pmatrix} \boldsymbol{\varsigma}^w_t \\ \boldsymbol{\varsigma}^a_t \end{pmatrix}. \tag{13}$$

In this article, the right invariant error is employed. Therefore, the system equation for the state error can be expressed as

$$\begin{pmatrix} \dot{\boldsymbol{\xi}}^r_{R_t} \\ \dot{\boldsymbol{\xi}}^r_{v_t} \\ \dot{\boldsymbol{\xi}}^r_{p_t} \\ \dot{\boldsymbol{\varsigma}}^w_t \\ \dot{\boldsymbol{\varsigma}}^a_t \end{pmatrix} = \hat{\mathbf{F}}_t \begin{pmatrix} \boldsymbol{\xi}^r_{R_t} \\ \boldsymbol{\xi}^r_{v_t} \\ \boldsymbol{\xi}^r_{p_t} \\ \boldsymbol{\varsigma}^w_t \\ \boldsymbol{\varsigma}^a_t \end{pmatrix} + \hat{\mathbf{G}}_t \begin{pmatrix} \mathbf{n}^w_t \\ \mathbf{n}^a_t \\ \mathbf{0}_{3\times 1} \\ \mathbf{n}^{b_w}_t \\ \mathbf{n}^{b_a}_t \end{pmatrix} \tag{14}$$

where $\hat{\mathbf{F}}_t$ denotes the state transition matrix, which is expressed as

$$\hat{\mathbf{F}}_t = \begin{pmatrix} \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & -\hat{\mathbf{R}}_t & \mathbf{0}_{3\times3} \\ [\mathbf{g}]_\times & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & -[\hat{\mathbf{v}}_t]_\times \hat{\mathbf{R}}_t & -\hat{\mathbf{R}}_t \\ \mathbf{0}_{3\times3} & \mathbf{I}_{3\times3} & \mathbf{0}_{3\times3} & -[\hat{\mathbf{p}}_t]_\times \hat{\mathbf{R}}_t & \mathbf{0}_{3\times3} \\ \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} \\ \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} \end{pmatrix} \tag{15}$$

where $\hat{\mathbf{G}}_t$ is the noise matrix, which is expressed as

$$\hat{\mathbf{G}}_t = \begin{pmatrix} \hat{\mathbf{R}}_t & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} \\ [\hat{\mathbf{v}}_t]_\times \hat{\mathbf{R}}_t & \hat{\mathbf{R}}_t & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} \\ [\hat{\mathbf{p}}_t]_\times \hat{\mathbf{R}}_t & \mathbf{0}_{3\times3} & \hat{\mathbf{R}}_t & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} \\ \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & -\mathbf{I}_{3\times3} & \mathbf{0}_{3\times3} \\ \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & -\mathbf{I}_{3\times3} \end{pmatrix}. \tag{16}$$

The propagation of the state's covariance matrix is expressed as

$$\hat{\Sigma}_t = \hat{\Phi}_t \hat{\Sigma}_{t-1} \hat{\Phi}_t^{\mathrm{T}} + \hat{\Phi}_t \hat{\mathbf{G}}_t \mathbf{Q} \hat{\mathbf{G}}_t^{\mathrm{T}} \hat{\Phi}_t^{\mathrm{T}} \Delta t \tag{17}$$

where Δt represents the time interval for IMU sampling data. $\mathbf{Q}$ is the noise covariance matrix, which is expressed as

$$\mathbf{Q} = \begin{pmatrix} \mathbf{q}_{11} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} \\ \mathbf{0}_{3\times3} & \mathbf{q}_{22} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} \\ \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} \\ \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{q}_{44} & \mathbf{0}_{3\times3} \\ \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{q}_{55} \end{pmatrix} \tag{18}$$

where

$$\begin{aligned} \mathbf{q}_{11} &= \mathrm{diag}\left(\left(\boldsymbol{\sigma}_t^w\right)^2\right) \\ \mathbf{q}_{22} &= \mathrm{diag}\left(\left(\boldsymbol{\sigma}_t^a\right)^2\right) \\ \mathbf{q}_{44} &= \mathrm{diag}\left(\left(\boldsymbol{\sigma}_t^{b_w}\right)^2\right) \\ \mathbf{q}_{55} &= \mathrm{diag}\left(\left(\boldsymbol{\sigma}_t^{b_a}\right)^2\right) \end{aligned} \tag{19}$$

$\hat{\boldsymbol{\Phi}}_t$ can be denoted as

$$\hat{\Phi}_t = \exp\left(\hat{\mathbf{F}}_t \Delta t\right). \tag{20}$$

B. State Update

In the proposed method, there are two primary measurements for state updates. The first one is the learning-based forward and vertical speeds, and the second one is the height constraint. In addition, the 3DIO can be extended with the aid of a behavior map, and a position update can also be applied. In this part, their measurement models are introduced.

1) Velocity Update: In Section IV, the forward speed v^{bf} and vertical speed v^{bv} are regressed from the IMU data using a neural network. Assuming the NHC, the lateral speed of the human can be considered zero. Since the velocity is formulated in the body frame, it needs to be transformed to the smartphone frame, denoted as

$$\tilde{\mathbf{z}}_t^v = \hat{\mathbf{R}}_b^s \cdot \begin{bmatrix} v^{bf} & 0 & v^{bv} \end{bmatrix}^{\mathrm{T}} \tag{21}$$

where $\hat{\mathbf{R}}_b^s$ denotes the estimated rotation matrix from the body frame to the smartphone frame; it is the transpose of $\hat{\mathbf{R}}_s^b$. The measurement model of the velocity can be expressed as

$$\tilde{\mathbf{z}}_t^v = \mathbf{R}_t^{\mathrm{T}} \mathbf{v}_t + \mathbf{n}_t^v \tag{22}$$

where $\mathbf{n}_t^v$ denotes the measurement noise. Equation (22) can be further expressed as

$$\begin{aligned} \tilde{\mathbf{Z}}_t^v &= \boldsymbol{\chi}_t^{-1} \mathbf{M}^v + \mathbf{N}_t^v \\ \begin{pmatrix} \tilde{\mathbf{z}}_t^v \\ -1 \\ 0 \end{pmatrix} &= \begin{pmatrix} \mathbf{R}_t^{\mathrm{T}} & -\mathbf{R}_t^{\mathrm{T}} \mathbf{v}_t & -\mathbf{R}_t^{\mathrm{T}} \mathbf{p}_t \\ \mathbf{0}_{1\times3} & 1 & 0 \\ \mathbf{0}_{1\times3} & 0 & 1 \end{pmatrix} \begin{pmatrix} \mathbf{0}_{3\times1} \\ -1 \\ 0 \end{pmatrix} + \begin{pmatrix} \mathbf{n}_t^v \\ 0 \\ 0 \end{pmatrix}. \end{aligned} \tag{23}$$

According to (23), the innovation can be expressed as

$$\mathbf{V}_t^v = \tilde{\mathbf{Z}}_t^v - \hat{\boldsymbol{\chi}}_t^{-1} \mathbf{M}^v. \tag{24}$$

The innovation can be further defined as

$$\begin{aligned} \mathbf{V}_t^v &= \hat{\boldsymbol{\chi}}_t \tilde{\mathbf{Z}}_t^v - \mathbf{M}^v \\ &= \begin{pmatrix} [\boldsymbol{\xi}_{R_t}^r]_\times & \boldsymbol{\xi}_{v_t}^r & \boldsymbol{\xi}_{p_t}^r \\ \mathbf{0}_{1\times3} & 0 & 0 \\ \mathbf{0}_{1\times3} & 0 & 0 \end{pmatrix} \begin{pmatrix} \mathbf{0}_{3\times1} \\ -1 \\ 0 \end{pmatrix} \\ &\quad + \begin{pmatrix} \hat{\mathbf{R}}_t & \hat{\mathbf{v}}_t & \hat{\mathbf{p}}_t \\ \mathbf{0}_{1\times3} & 1 & 0 \\ \mathbf{0}_{1\times3} & 0 & 1 \end{pmatrix} \begin{pmatrix} \mathbf{n}_t^v \\ 0 \\ 0 \end{pmatrix}. \end{aligned} \tag{25}$$

If only the first three rows are extracted, (25) can be expressed as

$$\mathbf{V}_t^v = -\begin{pmatrix} \mathbf{0}_{3\times3} & \mathbf{I}_{3\times3} & \mathbf{0}_{3\times3} \end{pmatrix} \begin{pmatrix} \boldsymbol{\xi}_{R_t}^r \\ \boldsymbol{\xi}_{v_t}^r \\ \boldsymbol{\xi}_{p_t}^r \end{pmatrix} + \hat{\mathbf{R}}_t \mathbf{n}_t^v. \tag{26}$$

Therefore, the measurement matrix of the velocity is denoted as

$$\mathbf{H}_t^v = \begin{pmatrix} \mathbf{0}_{3\times3} & \mathbf{I}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} \end{pmatrix}. \tag{27}$$

2) Position Update: The measurement model of the position can be expressed as

$$\tilde{\mathbf{z}}_t^p = \mathbf{p}_t + \mathbf{n}_t^p \tag{28}$$

where $\tilde{\mathbf{z}}_t^p$ and $\mathbf{n}_t^p$ are the position measurement and its noise, respectively. Equation (28) can be further expressed as

$$\begin{aligned} \tilde{\mathbf{Z}}_t^p &= \boldsymbol{\chi}_t \mathbf{M}^p + \mathbf{N}_t^p \\ \begin{pmatrix} \tilde{\mathbf{z}}_t^p \\ 0 \\ 1 \end{pmatrix} &= \begin{pmatrix} \mathbf{R}_t & \mathbf{v}_t & \mathbf{p}_t \\ \mathbf{0}_{1\times3} & 1 & 0 \\ \mathbf{0}_{1\times3} & 0 & 1 \end{pmatrix} \begin{pmatrix} \mathbf{0}_{3\times1} \\ 0 \\ 1 \end{pmatrix} + \begin{pmatrix} \mathbf{n}_t^p \\ 0 \\ 0 \end{pmatrix}. \end{aligned} \tag{29}$$

Therefore, the innovation can be written as

$$\begin{aligned} \mathbf{V}_t^p &= \hat{\boldsymbol{\chi}}_t^{-1} \tilde{\mathbf{Z}}_t^p - \mathbf{M}^p \\ &= \begin{pmatrix} \left[\boldsymbol{\xi}_{R_t}^l\right]_\times & \boldsymbol{\xi}_{v_t}^l & \boldsymbol{\xi}_{p_t}^l \\ \mathbf{0}_{1\times3} & 0 & 0 \\ \mathbf{0}_{1\times3} & 0 & 0 \end{pmatrix} \begin{pmatrix} \mathbf{0}_{3\times1} \\ 0 \\ -1 \end{pmatrix} \\ &\quad + \begin{pmatrix} \hat{\mathbf{R}}_t^{\mathrm{T}} & -\hat{\mathbf{R}}_t^{\mathrm{T}} \hat{\mathbf{v}}_t & -\hat{\mathbf{R}}_t^{\mathrm{T}} \hat{\mathbf{p}}_t \\ \mathbf{0}_{1\times3} & 1 & 0 \\ \mathbf{0}_{1\times3} & 0 & 1 \end{pmatrix} \begin{pmatrix} \mathbf{n}_t^p \\ 0 \\ 0 \end{pmatrix}. \end{aligned} \tag{30}$$

If only the first three rows are extracted, (30) can be expressed as

$$\mathbf{V}_t^p = -\begin{pmatrix}\mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{I}_{3\times3}\end{pmatrix}\begin{pmatrix}\boldsymbol{\xi}_{R_t}^l \\ \boldsymbol{\xi}_{v_t}^l \\ \boldsymbol{\xi}_{p_t}^l\end{pmatrix} + \hat{\mathbf{R}}_t^{\mathrm{T}}\mathbf{n}_t^p. \tag{31}$$

Therefore, the measurement matrix of the position is denoted as

$$\mathbf{H}_t^p = \begin{pmatrix}\mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{I}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3}\end{pmatrix}. \tag{32}$$

However, the above derivation is based on left invariant error. To ensure consistency, the left invariant error needs to be transformed into a right invariant error. Therefore, the new measurement matrix can be expressed as

$$\mathbf{H}_t^p = \mathbf{H}_t^p\begin{pmatrix}\mathbf{Ad}_{\hat{\chi}_t^{-1}} & \mathbf{0}_{9\times6} \\ \mathbf{0}_{6\times9} & \mathbf{I}_{6\times6}\end{pmatrix} \tag{33}$$

where

$$\mathbf{Ad}_{\hat{\chi}_t^{-1}} = \begin{pmatrix}\mathbf{E}_{11} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} \\ \mathbf{E}_{21} & \mathbf{E}_{11} & \mathbf{0}_{3\times3} \\ \mathbf{E}_{31} & \mathbf{0}_{3\times3} & \mathbf{E}_{11}\end{pmatrix} \tag{34}$$

where

$$\begin{aligned}\mathbf{E}_{11} &= \hat{\boldsymbol{\chi}}_{t\ 1:3,1:3}^{-1} \\ \mathbf{E}_{21} &= \left[\hat{\boldsymbol{\chi}}_{t\ 1:3,4}^{-1}\right]_\times \mathbf{E}_{11} \\ \mathbf{E}_{31} &= \left[\hat{\boldsymbol{\chi}}_{t\ 1:3,5}^{-1}\right]_\times \mathbf{E}_{11}\end{aligned} \tag{35}$$

where $\hat{\boldsymbol{\chi}}_{t\ i:j,i:j}^{-1}$ corresponds to the elements in $\hat{\boldsymbol{\chi}}_t^{-1}$ from rows i to j and columns i to j. $\hat{\chi}_{t\ i:j,i}^{-1}$ corresponds to the elements in $\hat{\boldsymbol{\chi}}_t^{-1}$ from rows i to j and at column i.

The height measurement is obtained by accumulating the regressed vertical speed, resulting in a scalar height. However, it can be observed that the equations above describe a general 3-D position update. If only a height constraint is available, we can determine the z-axis position of $\widetilde{\mathbf{z}}_t^p$, which cannot be directly integrated into the InEKF. To address this problem, the x-axis and y-axis components of $\widetilde{\mathbf{z}}_t^p$ can be set to zero. However, they are treated as pseudo measurements and considered invalid. This means that the first two elements on the main diagonal of the measurement covariance matrix are assigned large values. This method allows the scalar height to be used in the InEKF without being affected by pseudo-measurements.

The measurement matrices are used to calculate the filtering gains. Then, the state update can be represented as

$$\begin{aligned}\hat{\boldsymbol{\chi}}_t &= \exp\left(\mathrm{Ca}\left(\mathbf{K}^\xi\mathbf{V}\right)\right)\hat{\boldsymbol{\chi}}_t \\ \hat{\boldsymbol{\theta}}_t &= \hat{\boldsymbol{\theta}}_t + \mathbf{K}^\zeta\mathbf{V}\end{aligned} \tag{36}$$

where $\mathbf{K}^\xi$ and $\mathbf{K}^\zeta$ denote the filtering gains for the matrix Lie group and the bias, respectively. Their dimensions are 9×3 and 6×3. $\mathbf{V}$ consists of the first three rows of elements from $\hat{\boldsymbol{\chi}}_t\widetilde{\mathbf{Z}}_t$. $\widetilde{\mathbf{Z}}_t$ is a general symbol, depending on whether it is a velocity or height measurement. $\mathrm{Ca}(\mathbf{B})$ is a transformation, which can be expressed as

$$\mathrm{Ca}(\mathbf{B}) = \begin{pmatrix}[\mathbf{B}_{1:3}]_\times & \mathbf{B}_{4:6} & \mathbf{B}_{7:9} \\ \mathbf{0}_{2\times3} & \mathbf{0}_{2\times1} & \mathbf{0}_{2\times1}\end{pmatrix} \tag{37}$$

where $\mathbf{B}_{i:j}$ corresponds to the elements in $\mathbf{B}$ from rows i to j.

The state prediction and update described above form the core of the 3DIO. Additionally, the 3DIO can be enhanced with the aid of a behavior map to mitigate drift. In [33], we introduced an IMU-only SLAM, in which environment-induced behaviors, such as corner turnings, are used as landmarks. We have also extended the IMU-only mapping to a 3-D scenario that includes stair-taking behaviors as landmarks to create a 3-D behavior map. With this behavior map, we can employ map matching to calibrate the 3DIO using only IMU data. In map matching, corner-turning behaviors are not utilized as they can unnaturally restrict the user's motion. Instead, only stair-taking behaviors are detected through height changes. If the mean vertical speed within one second $\bar{v}^{bv}$ exceeds 0.1 m/s, based on our test results, we assume that the user has started stair-taking behaviors. Then, closest point matching is used to obtain absolute position information from the behavior map. Although closest point matching can easily lead to incorrect matches when the odometry has significant errors, the sparse and distant distribution of staircases helps mitigate such errors to some extent. It should be noted that the highest and lowest points of the stair will be chosen based on whether the user is descending or ascending, as shown below

$$\begin{cases}\widetilde{\mathbf{z}}_t^{\mathrm{lowp}} & \bar{v}^{bv} > 0.1 \text{ m/s} \\ \widetilde{\mathbf{z}}_t^{\mathrm{higp}} & \bar{v}^{bv} < -0.1 \text{ m/s}\end{cases} \tag{38}$$

where $\widetilde{\mathbf{z}}_t^{\mathrm{lowp}}$ and $\widetilde{\mathbf{z}}_t^{\mathrm{higp}}$ denote the lowest and highest positions of the corresponding stairs, respectively. Finally, the 3-D position can be incorporated into the InEKF using the position update to correct accumulated errors.

VI. Experimental Tests and Discussion

Two 3-D scenarios were selected as test sites to evaluate the performance of the proposed method: an office building and a multistory complex. An iPhone 12 was used to collect IMU data, while a Mid-360 sensor from Livox Technology Company was employed to run FAST-LIO [40], providing ground truth (GT) in the indoor environments. The experimental setup is illustrated in Fig. 3.

A. Evaluation Methods

In the tests, the following methods are used for comparison.

1) *3-D-PDR [41]:* This method locates the pedestrian using a model-based dead reckoning approach.
2) *RONIN [24]:* This method achieves localization by using a learning-based dead reckoning. There are three variants of RONIN, with the ResNet variant being used here.
3) *EKF-DIO:* This method combines inertial navigation with learned forward and vertical velocities, as well as NHC, within an EKF framework for state estimation.
4) *3DIO:* This is the proposed method that integrates inertial navigation, learned forward and vertical velocities, NHC, and height observation within an InEKF framework.
4) *3DIO (pos):* This method incorporates absolute positions, derived from matched stair-taking behaviors, into 3DIO to mitigate drift.

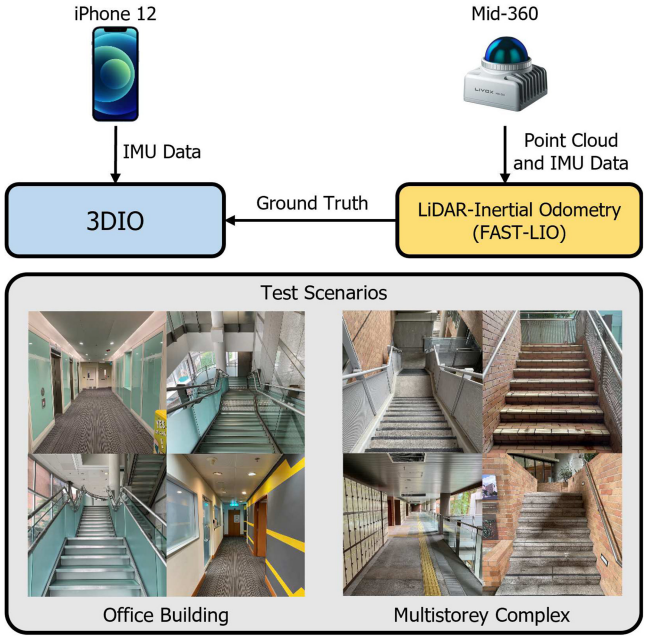

Fig. 3. Experimental scheme. A LiDAR-inertial integrated system was used to provide the GT. Test scenarios include the office building and multistorey complex.

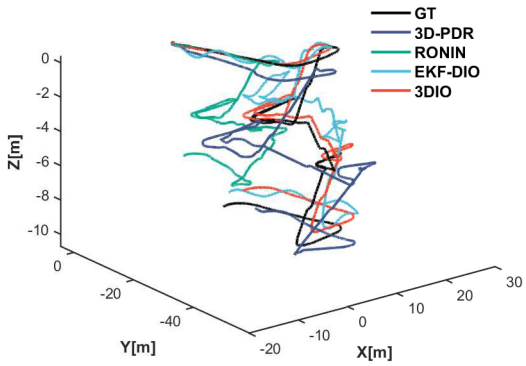

Fig. 4. Trajectory comparison among GT, 3-D-PDR, RONIN, EKF-DIO, and 3DIO in the office building.

B. Experiment in the Office Building

The evaluation was first conducted in the office building. The user carried both the smartphone and the Mid-360 while moving from the 3rd floor to the 1st floor, covering a distance of approximately 206 m.

The trajectory comparison is shown in Fig. 4, where GT denotes the GT provided by FAST-LIO. The error comparison for horizontal and vertical position estimations is presented in Fig. 5. The results demonstrate that the proposed method achieves the best positioning performance. In contrast, both 3-D-PDR and RONIN exhibit significant errors in horizontal and vertical positioning. First, 3-D-PDR is affected by heading uncertainty, leading to drift in horizontal position estimates. Additionally, accurately detecting stair-taking behavior using only a handheld smartphone proves challenging, which negatively impacts vertical position estimation. In the case of RONIN, attitude estimation operates as an open-loop system. The estimated attitude is influenced by unknown biases,

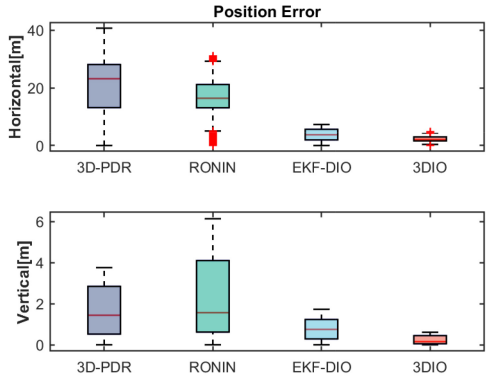

Fig. 5. Horizontal and vertical position error comparison among 3-D-PDR, RONIN, EKF-DIO, and 3DIO in the office building.

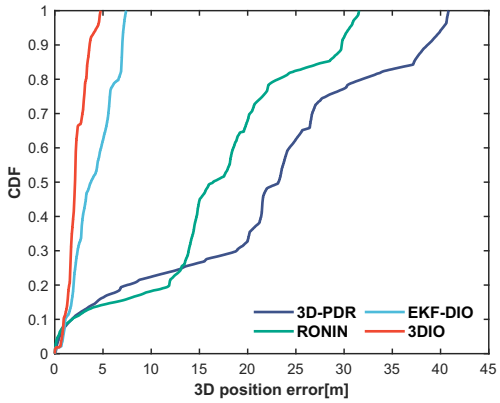

Fig. 6. 3-D position error CDF comparison among 3-D-PDR, RONIN, EKF-DIO, and 3DIO in the office building.

which affect the transformation of IMU data from the body frame to the world frame, degrading regression performance. The results indicate that EKF-DIO outperforms both 3-D-PDR and RONIN. In EKF-DIO, IMU bias can be estimated, resulting in better attitude correction and improved measurement regression performance. However, EKF is prone to linearization errors. Furthermore, the vertical channel in inertial navigation systems is inherently unstable [42], and vertical errors cannot be fully mitigated by vertical speed alone. As a result, EKF-DIO exhibits significant vertical position errors. The proposed 3DIO mitigates the impact of linearization errors. Additionally, the inclusion of a height constraint reduces vertical errors, making the proposed method superior to traditional approaches.

The 3-D position error cumulative distribution function (CDF) comparison is indicated in Fig. 6. The horizontal and vertical position absolute trajectory error (ATE) and relative trajectory error (RTE) comparison is shown in Table I. It shows that 3-D-PDR presents the largest error, with a total of 40.23 m (95%), including a horizontal ATE of 24.52 m and a vertical ATE of 2.13 m. The position errors of RONIN and EKF-DIO are 30.57 m (95%) and 7.14 m (95%). The position error of 3DIO is 4.36 m (95%) with a horizontal ATE of 2.47 m and a vertical ATE of 0.31 m.

TABLE I
HORIZONTAL AND VERTICAL POSITION ATE AND RTE COMPARISON AMONG 3-D-PDR, RONIN, EKF-DIO, AND 3DIO IN THE OFFICE BUILDING

Method	Horizontal		Vertical	
	ATE (m)	RTE (m)	ATE (m)	RTE (m)
3D-PDR	24.52	5.52	2.13	0.66
RONIN	18.38	4.43	3.49	0.78
EKF-DIO	4.44	1.73	0.91	0.43
3DIO	2.47	0.95	0.31	0.11

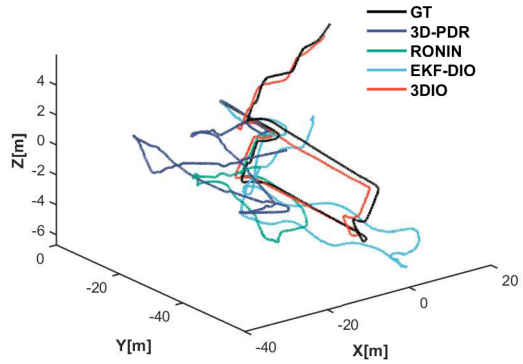

Fig. 7. Trajectory comparison among GT, 3-D-PDR, RONIN, EKF-DIO, and 3DIO in the multistorey complex.

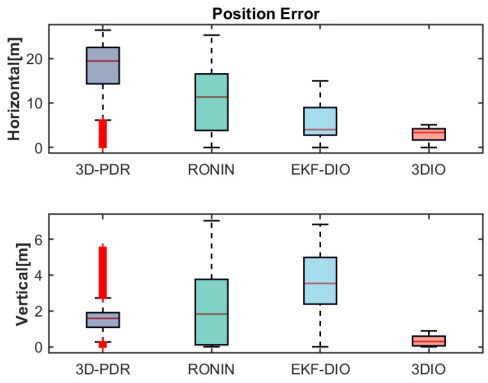

Fig. 8. Horizontal and vertical position error comparison among 3-D-PDR, RONIN, EKF-DIO, and 3DIO in the multistorey complex.

C. Experiment in the Multistorey Complex

Another field test was conducted in a multistorey complex to further evaluate the performance of the proposed method. The trajectory spans a distance of about 210 m.

The trajectory comparison is presented in Fig. 7. It demonstrates that traditional methods cannot accurately track the position, whereas the proposed 3DIO achieves trajectory estimation comparable to the GT from LiDAR-inertial odometry. The comparison of horizontal and vertical position errors is shown in Fig. 8. The results indicate that traditional methods exhibit larger errors in both horizontal and vertical positioning. Although EKF-DIO provides better horizontal position estimates than 3-D-PDR and RONIN, it remains inferior to 3DIO due to the impact of linearization errors. Furthermore, EKF-DIO still struggles with vertical position estimation. In contrast, the height constraint incorporated into 3DIO enables more accurate height estimation, outperforming EKF-DIO.

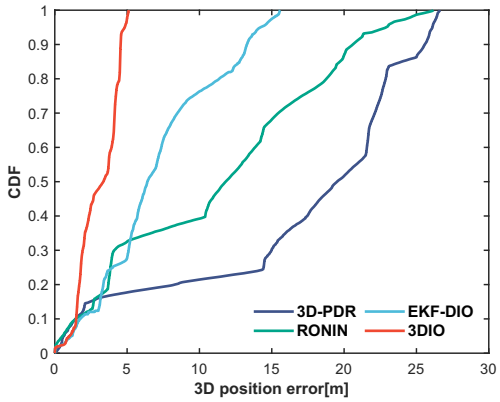

Fig. 9. 3-D position error CDF comparison among 3-D-PDR, RONIN, EKF-DIO, and 3DIO in the multistorey complex.

TABLE II
HORIZONTAL AND VERTICAL POSITION ATE AND RTE COMPARISON AMONG 3-D-PDR, RONIN, EKF-DIO, AND 3DIO IN THE MULTISTOREY COMPLEX

Method	Horizontal		Vertical	
	ATE (m)	RTE (m)	ATE (m)	RTE (m)
3D-PDR	18.47	5.71	2.35	1.02
RONIN	12.94	4.78	3.23	0.90
EKF-DIO	7.14	3.23	3.92	0.83
3DIO	3.30	0.90	0.44	0.10

The 3-D position error CDF comparison is given in Fig. 9. The horizontal and vertical position ATE and RTE is indicated in Table II. This demonstrates that the proposed 3DIO can achieve a lower position error compared to traditional methods. The position errors for 3-D-PDR, RONIN, and EKF-DIO are 26.11 m (95%), 22.90 m (95%), and 14.22 m (95%), respectively. All of them have an error greater than 10 m. The position error of the proposed 3DIO is 4.84 m (95%), with a horizontal ATE of 3.30 m and a vertical ATE of 0.44 m.

D. Discussion—Heading and Height Correction Performance

The improved performance of the 3DIO is due to the ability to correct the heading online and the introduction of a height constraint. Fig. 10 shows the heading comparison between the strapdown inertial navigation system (SINS) and the 3DIO. In SINS, the gyroscope is directly utilized to calculate the attitude through integration. It can be observed that the heading in SINS experiences drift because of unknown bias. However, in the 3DIO, the bias can be estimated and compensated for, resulting in better position estimation.

In this part, we also use a variable-controlling scheme to evaluate the effect of introducing the height constraint. We compare 3DIO-NH and 3DIO, with the vertical position error shown in Fig. 11. In 3DIO-NH, only the velocity observation is employed. It can be observed that the vertical channel in 3DIO-NH still experiences severe drift, as pure vertical speed cannot mitigate the error in the vertical channel of the inertial navigation system. In the proposed 3DIO, the height

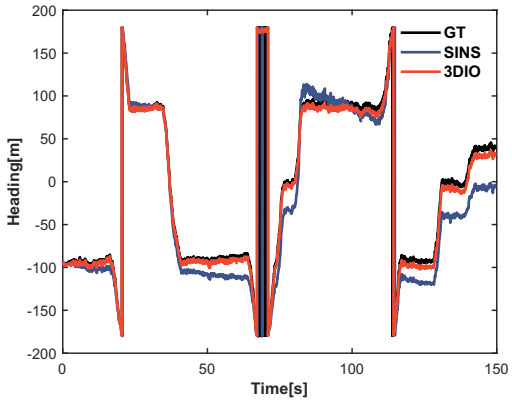

Fig. 10. Heading comparison between SINS and 3DIO.

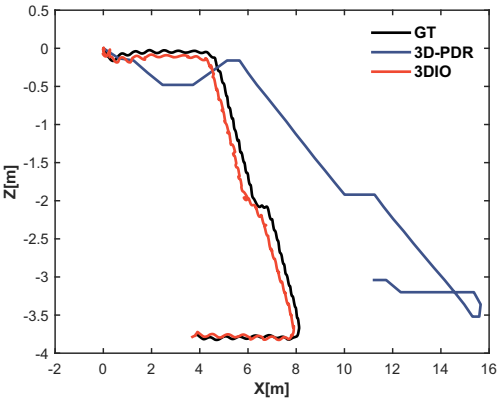

Fig. 12. Position comparison among GT, 3-D-PDR, and 3DIO.

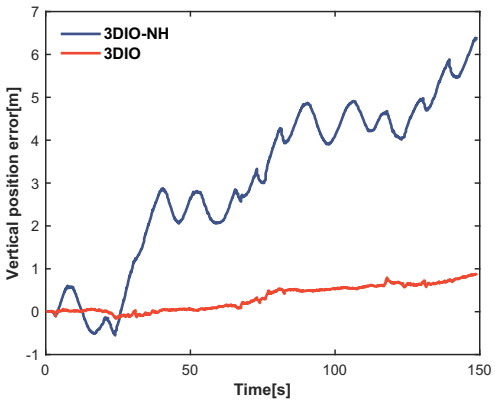

Fig. 11. Vertical position error comparison between 3DIO-NH and 3DIO.

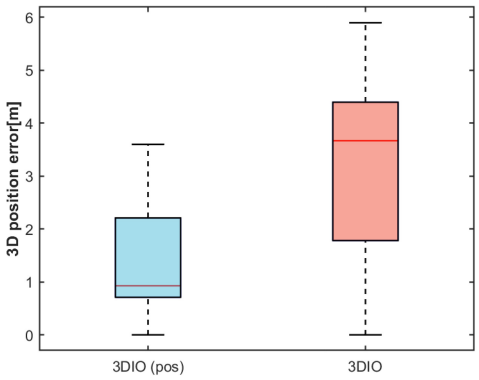

Fig. 13. 3-D position error comparison between 3DIO (pos) and 3DIO.

constraint is incorporated into the position update, effectively restraining error accumulation and achieving better vertical position estimation.

E. Discussion—Estimation Performance in Varying Motions

In real 3-D environments, users experience different types of motion. For example, when taking stairs, their movement is restricted by the stairs, resulting in a different step compared to walking on a flat surface. In traditional PDR methods, the user's step length is usually set as a fixed value, as these methods are designed for 2-D scenarios where the step length is relatively stable. However, when referring to taking stairs in 3-D scenarios, the estimated position will show significant errors if the step length remains fixed.

Fig. 12 presents a position comparison among GT, 3-D-PDR, and 3DIO. The figure provides an x-z view, illustrating both horizontal and vertical movement. According to the GT, the user starts moving on a flat surface from the point [0, 0], then descends the stairs, and finally continues on another flat surface. The blue line represents the position estimated by 3-D-PDR. First, 3-D-PDR fails to estimate height accurately; even on flat surfaces, it shows vertical position changes. This occurs because 3-D-PDR relies solely on raw IMU data to detect stair-taking behavior, making it prone to false detections due to noise in low-cost IMUs. Second, 3-D-PDR assumes a fixed step length, causing the estimated horizontal movement distance to appear longer when descending stairs. This is inaccurate, as the user's step length is typically shorter on stairs than on flat surfaces. In contrast, the proposed 3DIO achieves more accurate position estimation. It adapts to varying motions by leveraging the networks ability to capture data features and regress appropriate forward and vertical velocities, resulting in improved positioning performance in this scenario.

F. Discussion—Estimation Performance With Behavior Map

In the proposed 3DIO, only velocity and height measurements are used. While it can restrain the error accumulation of inertial navigation to some extent, it is essentially an odometry system, which faces drift in long-term applications. Therefore, a map can be introduced to calibrate the odometry error. To ensure this remains an IMU-only solution, environmental behaviors, such as corner turnings and stair takings, are used as landmarks. When the IMU detects these behaviors, corresponding positions from the map can be used to correct the 3DIO. However, in this article, we only use stair-taking behavior, which will not limit the user's movement unnaturally.

Fig. 13 indicates the 3-D position error comparison between 3DIO (pos) and 3DIO. Table III presents the 3-D position root mean square error (RMSE) comparison. In 3DIO (pos), absolute position from matched stair-taking behaviors is incorporated into the filter. With the aid of a behavior map, it has been shown that the position error of 3DIO can be reduced

TABLE III
3-D POSITION RMSE COMPARISON BETWEEN 3DIO (POS) AND 3DIO

Method	RMSE (m)
3DIO (pos)	1.76
3DIO	3.52

using absolute positioning data, resulting in better positioning accuracy.

VII. CONCLUSION

This article presents a low-drift 3-D deep-inertial odometry (3DIO) method for indoor pedestrian localization using an IMU. First, a deep learning-based approach is employed to regress forward and vertical speeds within the human body frame, which are combined with a NHC to generate velocity measurements. A height constraint is also applied, based on the regressed vertical speed, to reduce errors along the vertical axis. Subsequently, an InEKF-based fusion integrates inertial navigation, velocity, and height measurements for state estimation. Field tests are conducted across various environments to assess the performance of the proposed method. The results indicate that the proposed 3DIO effectively mitigates error drift in 3-D scenarios using only an IMU. Furthermore, we explore the method from multiple perspectives, including heading and height correction performance, the impact of varying motions, and the benefits provided by the behavior map. Our findings demonstrate that the proposed 3DIO offers significant improvements over traditional IMU-only localization methods.

In this work, we assume that the smartphone is held relatively static with respect to the human body, which presents a limitation of the proposed method. In the future, we plan to introduce an additional neural network to determine the relative pose between the smartphone and the pedestrian using IMU data, enabling real-time rotation compensation. Moreover, while this study relies solely on IMU data for indoor localization, we remain open to incorporating external data sources. For example, intermittent RF signals could be integrated to further constrain odometry errors, reducing the risk of incorrect matches with the behavior map and enhancing performance across diverse scenarios.

REFERENCES

[1] M. Rocha et al., "Indoor localization using fiber Bragg grating-based accelerometers for smart healthcare," *IEEE Trans. Consum. Electron.*, vol. 70, no. 1, pp. 68–77, Feb. 2024.

[2] X. Yi et al., "EgoLocate: Real-time motion capture, localization, and mapping with sparse body-mounted sensors," *ACM Trans. Graph.*, vol. 42, no. 4, pp. 1–17, 2023.

[3] P.-Y. Tseng, J. J. Lin, Y.-C. Chan, and A. Y. Chen, "Real-time indoor localization with visual SLAM for in-building emergency response," *Autom. Construct.*, vol. 140, Aug. 2022, Art. no. 104319.

[4] P. S. Farahsari, A. Farahzadi, J. Rezazadeh, and A. Bagheri, "A survey on indoor positioning systems for IoT-based applications," *IEEE Internet Things J.*, vol. 9, no. 10, pp. 7680–7699, May 2022.

[5] Y. Yu et al., "A novel 3-D indoor localization algorithm based on BLE and multiple sensors," *IEEE Internet Things J.*, vol. 8, no. 11, pp. 9359–9372, Jun. 2021.

[6] F. Zafari, A. Gkelias, and K. K. Leung, "A survey of indoor localization systems and technologies," *IEEE Commun. Surveys Tuts.*, vol. 21, no. 3, pp. 2568–2599, 3rd Quart., 2019.

[7] O. J. Woodman, "An introduction to inertial navigation," Univ. Cambridge Comput. Lab., Cambridge, U.K., Rep. UCAM-CL-TR-696, 2007.

[8] J. Lu, K. Chen, B. Li, and M. Dai, "Hybrid navigation method of INS/PDR based on action recognition," *IEEE Sensors J.*, vol. 18, no. 20, pp. 8541–8548, Oct. 2018.

[9] H. Weinberg, "Using the ADXL202 in pedometer and personal navigation applications," Application Note AN-602, Analog Devices, Wilmington, MA, USA, 2002.

[10] Y. Zhao, W. Gong, L. Li, B. Zhang, and C. Li, "An efficient and robust fingerprint-based localization method for multifloor indoor environment," *IEEE Internet Things J.*, vol. 11, no. 3, pp. 3927–3941, Feb. 2024.

[11] J. Dong, M. Noreikis, Y. Xiao, and A. Ylä-Jääski, "ViNav: A vision-based indoor navigation system for smartphones," *IEEE Trans. Mobile Comput.*, vol. 18, no. 6, pp. 1461–1475, Jun. 2019.

[12] Y. Dong, D. Yan, T. Li, M. Xia, and C. Shi, "Pedestrian gait information aided visual inertial SLAM for indoor positioning using handheld smartphones," *IEEE Sensors J.*, vol. 22, no. 20, pp. 19845–19857, Oct. 2022.

[13] S. Wen et al., "Enhanced pedestrian navigation on smartphones with VLP-assisted PDR integration," *IEEE Sensors J.*, vol. 23, no. 14, pp. 15952–15962, Jul. 2023.

[14] Q. Wang et al., "Recent advances in pedestrian inertial navigation based on smartphone: A review," *IEEE Sensors J.*, vol. 22, no. 23, pp. 22319–22343, Dec. 2022.

[15] Q. Wang et al., "Pedestrian dead reckoning based on walking pattern recognition and online magnetic fingerprint trajectory calibration," *IEEE Internet Things J.*, vol. 8, no. 3, pp. 2011–2026, Feb. 2021.

[16] L. E. Díez, A. Bahillo, J. Otegui, and T. Otim, "Step length estimation methods based on inertial sensors: A review," *IEEE Sensors J.*, vol. 18, no. 17, pp. 6908–6926, Sep. 2018.

[17] A. Martinelli, H. Gao, P. D. Groves, and S. Morosi, "Probabilistic context-aware step length estimation for pedestrian dead reckoning," *IEEE Sensors J.*, vol. 18, no. 4, pp. 1600–1611, Feb. 2018.

[18] Y. Yao, L. Pan, W. Fen, X. Xu, X. Liang, and X. Xu, "A robust step detection and stride length estimation for pedestrian dead reckoning using a smartphone," *IEEE Sensors J.*, vol. 20, no. 17, pp. 9685–9697, Sep. 2020.

[19] S. Yang, J. Liu, X. Gong, G. Huang, and Y. Bai, "A robust heading estimation solution for smartphone multisensor-integrated indoor positioning," *IEEE Internet Things J.*, vol. 8, no. 23, pp. 17186–17198, Dec. 2021.

[20] J. Choi and Y.-S. Choi, "Calibration-free positioning technique using Wi-Fi ranging and built-in sensors of mobile devices," *IEEE Internet Things J.*, vol. 8, no. 1, pp. 541–554, Jan. 2020.

[21] C. Chen, X. Lu, A. Markham, and N. Trigoni, "IONet: Learning to cure the curse of drift in inertial odometry," in *Proc. AAAI Conf. Artif. Intell.*, 2018, pp. 6468–6476.

[22] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, 1997.

[23] H. Yan, Q. Shan, and Y. Furukawa, "RIDI: Robust IMU double integration," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2018, pp. 621–636.

[24] S. Herath, H. Yan, and Y. Furukawa, "RONIN: Robust neural inertial navigation in the wild: Benchmark, evaluations, & new methods," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, 2020, pp. 3146–3152.

[25] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 770–778.

[26] S. Bai, J. Z. Kolter, and V. Koltun, "An empirical evaluation of generic convolutional and recurrent networks for sequence modeling," 2018, *arXiv:1803.01271*.

[27] Y. Wang, H. Cheng, C. Wang, and M. Q.-H. Meng, "Pose-invariant inertial odometry for pedestrian localization," *IEEE Trans. Instrum. Meas.*, vol. 70, pp. 1–12, Jul. 2021.

[28] X. Cao, C. Zhou, D. Zeng, and Y. Wang, "RIO: Rotation-equivariance supervised learning of robust inertial odometry," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2022, pp. 6614–6623.

[29] W. Liu et al., "TLIO: Tight learned inertial odometry," *IEEE Robot. Autom. Lett.*, vol. 5, no. 4, pp. 5653–5660, Oct. 2020.

[30] Y. Wang, J. Kuang, X. Niu, and J. Liu, "LLIO: Lightweight learned inertial odometer," *IEEE Internet Things J.*, vol. 10, no. 3, pp. 2508–2518, Feb. 2023.

[31] T. Qin, P. Li, and S. Shen, "Relocalization, global optimization and map merging for monocular visual-inertial SLAM," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, 2018, pp. 1197–1204.
[32] Y. Zhong, W. Wen, and L.-T. Hsu, "Trajectory smoothing using GNSS/PDR integration via factor graph optimization in urban canyons," *IEEE Internet Things J.*, vol. 11, no. 14, pp. 25425–25439, Jul. 2024.
[33] S. Bai, W. Wen, L.-T. Hsu, and P. Yang, "Factor graph optimization-based smartphone IMU-only indoor SLAM with multihypothesis turning behavior loop closures," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 60, no. 6, pp. 8380–8400, Dec. 2024.
[34] W. Wen, X. Bai, Y. C. Kan, and L.-T. Hsu, "Tightly coupled GNSS/INS integration via factor graph and aided by fish-eye camera," *IEEE Trans. Veh. Technol.*, vol. 68, no. 11, pp. 10651–10662, Nov. 2019.
[35] A. Barrau and S. Bonnabel, "The invariant extended Kalman filter as a stable observer," *IEEE Trans. Autom. Control*, vol. 62, no. 4, pp. 1797–1812, Apr. 2017.
[36] M. Brossard, A. Barrau, and S. Bonnabel, "RINS-W: Robust inertial navigation system on wheels," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, 2019, pp. 2068–2075.
[37] R. Hartley, M. Ghaffari, R. M. Eustice, and J. W. Grizzle, "Contact-aided invariant extended Kalman filtering for robot state estimation," *Int. J. Robot. Res.*, vol. 39, no. 4, pp. 402–430, 2020.
[38] E. R. Potokar, K. Norman, and J. G. Mangelson, "Invariant extended Kalman filtering for underwater navigation," *IEEE Robot. Autom. Lett.*, vol. 6, no. 3, pp. 5792–5799, Jul. 2021.
[39] D. P. Kingma, "Adam: A method for stochastic optimization," 2014, *arXiv:1412.6980*.
[40] W. Xu and F. Zhang, "Fast-LIO: A fast, robust LiDAR-inertial odometry package by tightly-coupled iterated Kalman filter," *IEEE Robot. Autom. Lett.*, vol. 6, no. 2, pp. 3317–3324, Apr. 2021.
[41] K. Itzik and L. Yaakov, "Step-length estimation during movement on stairs," in *Proc. 27th Mediterr. Conf. Control Autom. (MED)*, 2019, pp. 518–523.
[42] V. Chueh, T.-C. Li, and R. Grethel, "INS/baro vertical channel performance using improved pressure altitude as a reference," in *Proc. IEEE/ION Posit., Locat. Navig. Symp.*, 2008, pp. 1199–1202.

Shiyu Bai (Member, IEEE) was born in Xuzhou, Jiangsu, China. He received the Ph.D. degree in navigation, guidance and control from Nanjing University of Aeronautics and Astronautics, Nanjing, China, in 2022.

He is currently a Postdoctoral Fellow with the Department of Aeronautical and Aviation Engineering, The Hong Kong Polytechnic University, Hong Kong. His research interests include inertial navigation, multisensor fusion, indoor positioning, and vehicular positioning.

Weisong Wen (Member, IEEE) received the B.Eng. degree in mechanical engineering from Beijing Information Science and Technology University, Beijing, China, in 2015, the M.Eng. degree in mechanical engineering from China Agricultural University, Beijing, China, in 2017, and the Ph.D. degree in mechanical engineering from The Hong Kong Polytechnic University (PolyU), Hong Kong, in 2020.

He was also a visiting Ph.D. student with the Faculty of Engineering, University of California, Berkeley, Berkeley, CA, USA, in 2018. Before joining PolyU as an Assistant Professor in 2023, he has been a Research Assistant Professor with AAE, PolyU since 2021. His research interests include the trustworthy multisensory integration, LiDAR SLAM, GNSS positioning, and autonomous systems.

Chuang Shi received the Ph.D. degree from Wuhan Institute of Surveying and Mapping (current Wuhan University), Wuhan, China, in 1998.

He currently serves as a Professor with the School of Electronic and Information Engineering and the Research Institute for Frontier Science, Beihang University, Beijing, China. Additionally, he also holds the position of Director with the Key Laboratory of Satellite Navigation and Mobile Communication Fusion Technology, a key laboratory under the Ministry of Industry and Information Technology, Beijing. His research interests include network adjustment, precise orbit determination of GNSS satellites and LEOs, as well as indoor positioning.