An Inertial Odometry and Enhanced Occupancy Grid Inertial SLAM for Legged Robots

Xiaodong Li, Zhi Xiong, Lipo Wang, *Senior Member, IEEE*, and Yan Cui

***Abstract*—Accurate position estimation is crucial for legged robots, the inertial odometry based on inertial measurement units (IMUs) is low-cost and easy to deploy, and inertial simultaneous localization and mapping (SLAM) methods can maintain stable performance on different robot platforms and degraded environments. In this article, a single IMU is mounted on the robot foot to construct the inertial odometry without relying on kinematic modeling. The stance phase and optimal zero-velocity point (OZP) are detected through IMU outputs, and the once zero-velocity update is performed at each OZP to correct the drift of inertial odometry. Furthermore, the enhanced occupancy grid inertial SLAM framework is introduced. In the grid map update section, the motion control vectors derived from the inertial odometry are used with the Bresenham algorithm to calculate accessibility probabilities of grid cells, which addresses detail loss from grid map discretization and the imbalance in the accessibility probability resulting from repeated erroneous map updates. Experimental tests on the Unitree B1 robot demonstrate that our system provides further improvement in position estimation.**

***Index Terms*—Inertial measurement unit (IMU), legged robot, odometry, simultaneous localization and mapping (SLAM).**

I. Introduction

With the rapid development of the Internet of Things (IoT) [1], [2] area, location-based services (LBS) technology for robots [3], [4] has become a key component. In complex applications, such as warehouse, inspection, and search operations, legged robots [5], [6] are increasingly prominent in robotics research due to their exceptional obstacle-crossing capabilities and flexibility. However, compared to traditional wheeled robots, legged robots have more complex motion patterns and structures. Accurate state and environment estimation is required for reliable control, motion planning, and decision-making for legged robots. Consequently, designing a low-cost and general positioning and navigation system for legged robots has become a significant research focus within the IoT field [7], [8].

Applications of legged robots typically include warehouse automation and building inspections. In indoor environments, the global navigation satellite system (GNSS) [9] often suffers from severe signal attenuation or reflection due to buildings and walls, significantly reducing the positioning accuracy. Furthermore, anchor-based localization methods, such as Wi-Fi [10], Bluetooth [11], and ultra-wideband (UWB) [12], face multiple challenges. These systems typically require multiple base stations to be installed in the environment to ensure full coverage of the localization area, which complicates the initial setup and increases equipment costs. In addition, regular maintenance and calibration limit widespread adoption in practical applications. In contrast, autonomous localization relying on self-configured sensors becomes a solution with great potential.

One of the most widely applicable methods for obtaining the state of legged robots is the leg odometry (LO) [13], [14]. This approach employs proprioceptive sensors, such as joint encoders and contact sensors, to infer the state of robots. With the assumption that the contact foot remains stationary, it leverages forward kinematics to estimate the robot's position and attitude. LO has been successfully deployed in various legged robot platforms. Lin et al. [15] proposed a LO method for hexapod robots to compute instantaneous estimates of body pose, but it requires the robot be supported by three legs. This significantly restricts agile movements and requires further consideration for adaptability to varied terrains. Inertial measurement units (IMUs) capture the orientation and displacement of a target, and researchers have investigated integrating the IMU into legged robots to improve state estimation accuracy. A notable method by Bloesch et al. [16] provided a versatile state estimation approach for different types of legged robots. This approach integrated foot position into the state vector, performs state prediction using the IMU outputs, and updates the state estimate during the stand phase based on the relative position from the foot to the base. A later improvement for humanoid robots introduced a key enhancement: incorporating foot orientation into the state vector, further improving estimation accuracy [17]. Agrawal et al. [18] proposed a state estimator based on a factor graph that enhances accuracy and robustness by combining preintegrated IMU measurements, forward kinematics, and contact detection with proprioceptive inertial data. Besides, learning-based approaches to LO have attracted substantial attention. Buchanan et al. [19] developed a proprioceptive

Received 19 November 2024; revised 21 March 2025 and 9 May 2025; accepted 5 June 2025. Date of publication 11 June 2025; date of current version 25 August 2025. This work was supported in part by the National Natural Science Foundation of China under Grant 62073163; in part by the Fundamental Research Funds for the Central Universities under Grant QZPY202310; and in part by the Jiangsu Graduate Research and Innovation Plan Project under Grant KYCX23_0384. *(Corresponding authors: Zhi Xiong; Lipo Wang.)*

Xiaodong Li, Zhi Xiong, and Yan Cui are with the College of Automation Engineering, Nanjing University of Aeronautics and Astronautics, Nanjing 211106, Jiangsu, China (e-mail: xiaodongli@nuaa.edu.cn; xiongzhi@nuaa.edu.cn; cuiyan@nuaa.edu.cn).

Lipo Wang is with the School of Electrical and Electronic Engineering, Nanyang Technological University, Singapore 639798 (e-mail: elpwang@ntu.edu.sg).

Digital Object Identifier 10.1109/JIOT.2025.3578665

state estimation method for legged robots that leverages IMU data to learn displacement measurements, the learned displacements are integrated with traditional LO, significantly reducing estimation drift over challenging terrains. Yang et al. [20] combined IMU and kinematic data fusion by employing neural networks to learn weight functions for foot force and LO states, improving observation accuracy in invariant extended Kalman filter. Nonetheless, a significant limitation of learning-based approaches is their reliance on large amounts of labeled datasets. To further reduce position drift and leverage environmental data, some legged robots incorporate external sensors, such as cameras and light detection and ranging (LiDAR), with LO. Cerberus [21] is a factor graph-based framework for visual-inertial-LO that enables online estimation of kinematic parameters, but its reliance on visual information may lead to performance degradation in visually challenging environments. Wisth et al. [22] proposed the VILENS method, which combines preintegrated velocity factors with visual, LiDAR, and IMU outputs to estimate a linear velocity bias term. This bias term was then incorporated into the system state to minimize the odometry drift. Although the above approach partially bounds the drift of robot state estimators, localization methods that fuse LO remain vulnerable to encoder noise and inaccuracies in kinematic modeling [23]. Moreover, minor mechanical deformations can introduce errors in velocity estimation, resulting in drift in position and orientation over time. Reliable ground contact detection [24] is essential for LO and is typically based on the output of foot force sensors. However, force sensors can be costly and prone to damage, which may lead to false detections or missed contacts. Moreover, designing effective LO systems requires an accurate and predefined kinematic model of the robot. However, obtaining precise model parameters can be challenging in some cases, limiting the universality of this approach. Additionally, LO becomes ineffective if foot slippages occur. In contrast, the IMU-based inertial odometry is cost-effective and easy to configure. It is already widely implemented in the pedestrian navigation area, such as the zero-velocity update (ZUPT) method [25]. Legged robots have similar motion characteristics to humans, making inertial odometry a promising solution for position estimation in these robots.

Simultaneous localization and mapping (SLAM) methods enable robots to build maps of unknown environments while determining their own poses. The ORB-SLAM series [26], [27], [28] represents a pivotal advancement in visual SLAM, utilizing features to build the map and estimate the camera pose, which is computationally efficient and resilient to rotation and scale changes. LiDAR-based SLAM methods rely on distance measurements. Fast-LIO [29] employs a tightly coupled approach to integrate raw point clouds with IMU measurements. This method achieves high accuracy while significantly reducing computational load. Recently, learning-based SLAM methods [30] have risen to prominence, differentiable SLAM-Net [31] embeds a particle filter-based SLAM into a differentiable neural network, enabling end-to-end learning for robust visual navigation. DLC-SLAM [32] enhances LiDAR-based SLAM with deep learning-based point cloud denoising and efficient loop closure, improving localization in challenging environments. However, learning-based SLAM requires large datasets for training and may struggle with generalization in unseen environments. Its higher computational cost can limit real performance. Despite significant advances in vision and LiDAR-based SLAM, these methods face several application challenges. First, the high cost of sensors and the complexities involved in sensor calibration present limitations for large-scale applications. Second, the performance of SLAM systems is often severely impacted by environmental conditions. Challenges arise in dynamic, cluttered, or featureless environments, leading to inaccuracies in overall estimation.

Inertial SLAM [33] shows the advantages as a potential solution, it obtains target state information and senses environmental features simultaneously via inertial odometry. This approach is especially well-suited for resource-constrained robotic platforms and can be applied in a broad range of environments and terrains. FootSLAM [34], the first generation of inertial SLAM, was designed for foot-mounted pedestrian navigation systems [25]. FootSLAM is based on Bayesian [35] and particle filtering [36], which divides the space into multiple hexagonal grids and corrects the cumulative error by updating the grid map. However, computational complexity increases dramatically as the mapping region expands. To overcome this limitation, a data structure known as the H-tree [37] was proposed that offers a compact representation of the hexagonal map. This structure reduces computational complexity from linear to logarithmic growth, enabling the FootSLAM algorithm to function in real-time systems. Subsequently, PlaceSLAM [38] and MagSLAM [39] were proposed, focusing on the association of stabilized environmental characteristics with maps to improve the accuracy of the localization. ActionSLAM [40] recognized specific pedestrian actions (such as opening doors, climbing stairs, and jumping), limiting system drift through the correlation between actions and locations. However, system stability is challenging to maintain because of the variability of human actions and their weak correlation with geolocations.

In our previous study, the occupancy grid inertial SLAM (OGI-SLAM) [41] used a square grid map to represent the environment. Unlike FootSLAM, which calculates the probability for each edge of a hexagonal grid, OGI-SLAM only updates the accessibility probability for each grid cell, providing a significant advantage in computational efficiency. Building on it, the distributed collaborative inertial SLAM [42] framework was introduced. This framework meets the requirements for cooperative localization without relative measurements and resolves the initial pose alignment issue. Additionally, OGI-SLAM2 [43] that includes a global optimization step was developed, it extracts key points from the trajectory and establishes associations among them. Through bundle adjustment, localization errors at key points are corrected, with the correction between common points modeled as an affine transformation. Currently, the OGI-SLAM series has shown strong performance in pedestrian navigation. However, significant challenges remain for adapting these methods to legged robotics, mainly, including:

1) Lack of a stable inertial odometry system, which is necessary to provide reliable front-end input for OGI-SLAM; 2) Detail loss due to grid map discretization and incorrect particle updates due to the imbalance in the accessibility probability of grid maps.

This article aims to establish an inertial odometry and enhanced OGI-SLAM (EOGI-SLAM) for legged robots. The proposed system consists of a single IMU and is suitable for various types of legged robots. It can function as a standalone navigation system, with the potential to integrate additional sensor data. The main contributions of this study are as follows.

1) An inertial odometry method relying solely on a single IMU is proposed for various legged robot platforms. The proposed odometry is designed to operate independently of kinematic models and without assumptions about gait or terrain, providing a simple and cost-effective solution.
2) An EOGI-SLAM framework is proposed that utilizes motion vectors and the Bresenham algorithm to determine the accessibility probabilities of grid cells. The framework avoids map discretization and imbalance, thus improving the overall performance of the SLAM system.
3) The field experiment conducted on the Unitree B1 robot demonstrates that the proposed inertial odometry exhibits better accuracy than traditional LO. The proposed EOGI-SLAM system more accurately captures grid map properties, maintains the continuity and global consistency of the map, and enhances overall localization performance.

The remainder of the article is organized as follows. Section II introduces the preliminaries and challenges for conventional OGI-SLAM. Section III provides an overview of the proposed system and details the algorithm in two parts. Then, Section IV evaluates the system's performance, and Section V concludes the article.

II. Preliminaries and Challenges Description

A. Preliminaries of OGI-SLAM

The OGI-SLAM is built on the Fast-SLAM [44] framework, which decomposes the state estimation problem into several low-dimensional estimation problems. In this approach, the Rao–Blackwellized particle filter (RBPF) [45] estimates the target's pose, while the environmental map is independently estimated by filters within each particle. Due to its robust adaptability to nonlinear motion models and its exceptional real-time performance, Fast-SLAM is an excellent solution for the estimation of robot state. As with all SLAM approaches, the core of OGI-SLAM lies in estimating the joint probability distribution of the target's pose and the environment map:

$$p(\mathbf{s}_{1:t}, \Theta \mid \mathbf{z}_{1:t}, \mathbf{u}_{1:t}) \tag{1}$$

where $\mathbf{s}_{1:t}$ represents the target's pose, and Θ denotes the environment map. The control vector $\mathbf{u}_{1:t}$ represents the motion vector based on odometry estimation, and $\mathbf{z}_{1:t}$ represents the map observations.

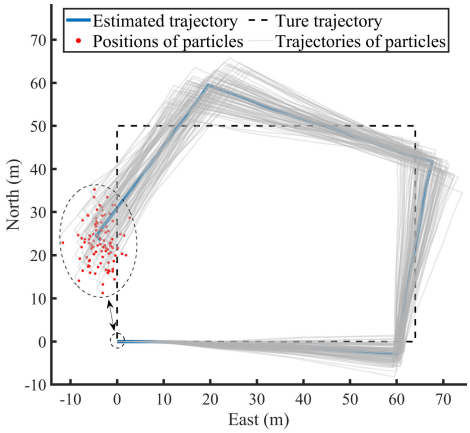


Fig. 1. OGI-SLAM system fails to achieve correct loop closure due to the low accuracy of the odometry input.

OGI-SLAM employs the grid map model, and the entire map Θ is divided into several equally sized grid cells θ. As the target moves through the space, the accessibility probability of grid cells is updated based on the current pose. Specifically, the process involves first determining the index of the grid cell occupied by the particle

$$\mathbf{p}^{\theta} = \left\langle \left[p^x / l^{\theta}, p^y / l^{\theta} \right]^T \right\rangle \tag{2}$$

where l^{θ} is the grid length of each cell, $[p^x, p^y]^T$ is the particle's current position, and $\langle \cdot \rangle$ denotes rounding each element of the vector.

Then, within the framework of the binary Bayesian filter [35], the accessibility probability of the current grid cell is updated based on the number of times it is occupied. Subsequently, particle weight updates are performed to obtain the posterior estimate of the target's pose and the environment map.

B. Challenges in OGI-SLAM

The inputs to OGI-SLAM come from the odometry, and the accuracy of the odometry estimates directly impacts the performance of the system. Significant drift of odometry outputs can cause particles in the pose prediction step to fail to cover the correct position. Furthermore, conventional map update methods lead to map dispersion and the imbalance of accessibility probability, which hinder effective loop closure or result in incorrect loop closures, ultimately leading to system failure.

Fig. 1 depicts a scenario where the SLAM system fails to close the loop correctly due to low odometry input accuracy. As shown, the particles gradually deviate from the true trajectory during the prediction phase due to significant odometry drift. The dashed box shows the range of the particle position after several predictions after several iterations. The system fails to establish an effective loop closure at the expected point, causing the trajectory to diverge from the true path. This highlights the importance of developing a reliable front-end odometry, as improving the accuracy of the inputs is crucial for ensuring the stability of the OGI-SLAM system.

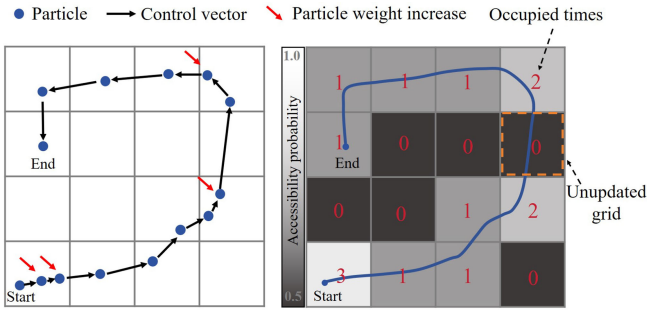


Fig. 2. Map update method in OGI-SLAM results in discrete and unbalanced maps.

On the other hand, the map update method in OGI-SLAM can result in discrete and unbalanced maps. Fig. 2 illustrates this challenge, where blue dots represent the positions of particles at different time steps, black arrows indicate the control vectors derived from odometry, and the different colors of the grid indicate the accessibility probabilities. It shows that the same grid cell is repeatedly occupied when the robot's step length is small, and conversely, when the step length increases, particles may cross certain cells without updating, leading to a discretization of the grid map, as highlighted by the yellow dashed box. Certain grid cells are updated repeatedly, while others traversed lack necessary updates. This tends to cause erroneous updating of particle weights and reduces the accuracy of loop closure.

III. Methodology

The overview of the proposed algorithm is shown in Fig. 3. The algorithm in this article consists of two main components: 1) the front-end is an inertial odometry for legged robots and 2) the back-end is an EOGI-SLAM framework. The proposed inertial odometry is primarily designed by placing an IMU on the robot's foot. The IMU signals are utilized to detect the periodic swing and stance phases of the robot's legs. In each stance phase, the optimal zero-velocity point (OZP) is identified. An error state Kalman filter (ESKF) is then designed and the once zero-velocity update (OZUPT) at the OZP is performed to correct the odometry drift. The main advantage of the proposed inertial odometry is simple deployment and independent of the environmental prior information or the robot's kinematic model. In the EOGI-SLAM method, the main steps include pose prediction, grid map update, particle weight update, and resampling. For the motion vector-driven grid map update method, the accessibility probabilities of grids are updated based on the extent to which the motion vector traverses each grid cell without altering the map resolution and increasing map memory. This approach prevents incorrect repetitive updates to the grid, avoiding erroneous particle loop closure. Additionally, it ensures continuity and balance of the grid map, thereby maintaining the global consistency of the system.

A. Inertial Odometry of Legged Robots

1) Gait Analayse and OZP Identification: During the movement of a legged robot, each leg transitions regularly between the swing and stance phases. The robot achieves locomotion by continuously alternating between these two phases. Fig. 4 illustrates the gait transition process during the robot's movement, t_1 and t_2 represent the swing phase, during which the foot lifts off the ground and swings. The moment from t_3 to t_6 corresponds to the stance phase, where the foot stops swinging and contacts the ground, providing stable support for the body. The robot then uses the contact point as a pivot to propel the body, achieving stable step displacement.

The indoor optical motion capture equipment is used to analyze the foot velocity characteristics of legged robots, and the results are shown in Fig. 5. The shaded areas represent the stance phase, during which the foot velocity approaches zero. Due to slight rotations of the foot upon ground contact, the foot velocity is not exactly zero. Given the distinct velocity characteristics during the stance phase, an IMU is placed near the robot's foot to construct the proposed inertial odometry. Notably, the minimum velocity point within each stance phase, referred to as the OZP, is observed to be even closer to zero. The statistics of 50 gait cycles show the velocity during the stance phase is 0.0558 m/s, while the average velocity at OZPs is measured as 0.03396 m/s. Therefore, applying the zero-velocity observation exclusively at OZPs can effectively reduce velocity observation errors caused by slight rotations and displacements of the IMU during the stance phase.

Accurately distinguishing stance phases and identifying OZPs are critical steps in the proposed odometry. The phase detection model is constructed using the adaptive generalized likelihood ratio test (GLRT) method [46]. The IMU signals are processed within a sliding window to obtain the detection statistic $T(n)$ at moment n

$$T(n) = \frac{1}{r}\sum_{k=n-r/2}^{n+r/2}\left(\frac{1}{\sigma_a^2}\left\|\boldsymbol{a}_k^b - \boldsymbol{g}^n\frac{\bar{\boldsymbol{a}}_k^b}{\left\|\bar{\boldsymbol{a}}_k^b\right\|}\right\|^2 + \frac{1}{\sigma_w^2}\left\|\boldsymbol{w}_k^b\right\|^2\right) \quad (3)$$

where r is the size of the sliding window, $\boldsymbol{a}_k^b$ and $\boldsymbol{w}_k^b$ present the acceleration and angular rate under body frame (b-frame) at time k, σ_a^2 and σ_w^2 denote the noise variance of them, respectively. $\boldsymbol{g}^n$ is the gravity vector in the navigation frame (n-frame), and $\|\cdot\|$ denotes the norm, which measures the magnitude of a vector.

When the detection statistic T is less than the adaptive threshold γ, which is defined in detail in [46], the moment is considered as the stance phase. The point with the smallest detection statistic within each detected stance phase is considered the OZP

$$t_{\text{ozp}} = \min_{n\in\mathcal{T}_{\text{stance}}} T(n) \quad (4)$$

where $\mathcal{T}_{\text{stance}}$ denotes an independent stance phase.

2) System and Observation Model: The inertial odometry developed for legged robots in this article is based on the strapdown inertial navigation system (INS). Given the significant bias and noise of low-cost IMUs, minor error terms in the state updates are neglected to reduce the complexity and enhance the general applicability of the algorithm. The

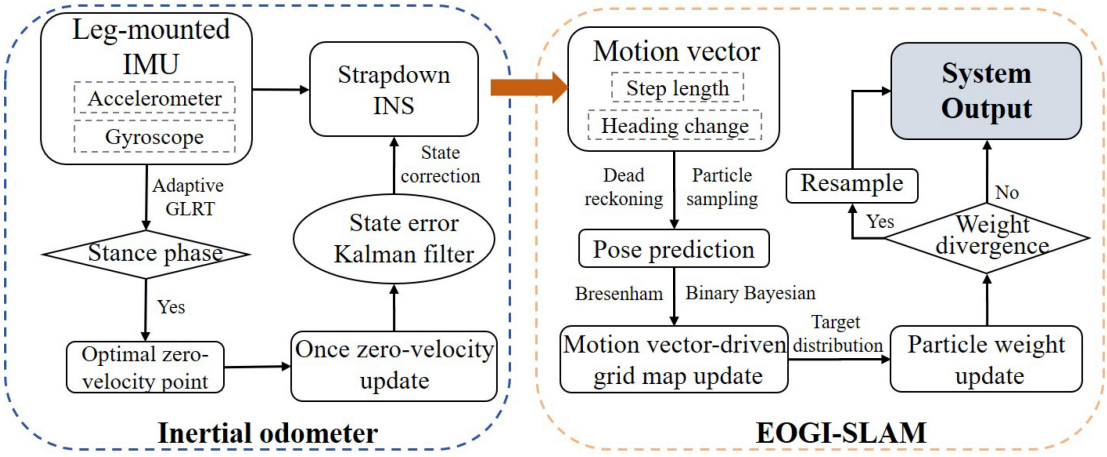


Fig. 3. Overview of the proposed system, consisting of two main components: front-end inertial odometry and back-end EOGI-SLAM optimization framework.

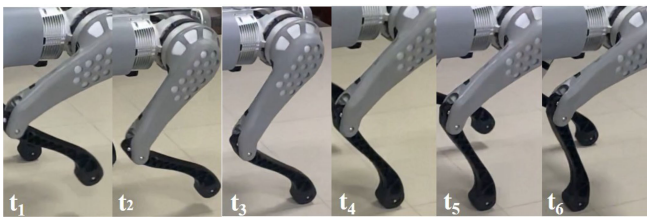


Fig. 4. Transitions between stance phases and swing phases during legged robot motions.

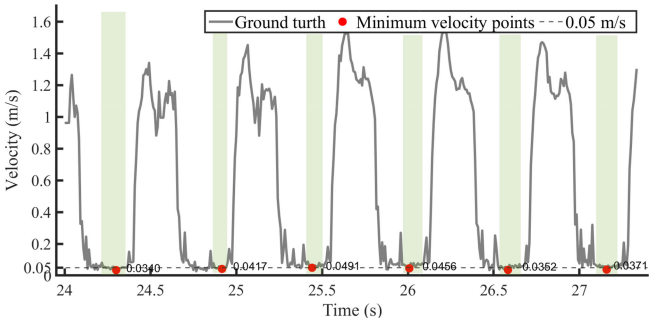


Fig. 5. Foot velocity characteristics of legged robots during motion.

simplified equations for attitude, velocity, and position updates are as follows:

$$\begin{cases} \dot{\boldsymbol{q}}_b^n = \frac{1}{2}\boldsymbol{\Omega}(\boldsymbol{w}^b)\cdot\boldsymbol{q}_b^n \\ \dot{\boldsymbol{v}}^n = \boldsymbol{C}_b^n\boldsymbol{f}^b - 2\boldsymbol{w}_{ie}^n\times\boldsymbol{v}^n + \boldsymbol{g}^n \\ \dot{\boldsymbol{p}}^n = \boldsymbol{v}^n \end{cases} \tag{5}$$

where $\boldsymbol{q}_b^n$ is the quaternion that defines the transformation from the b-frame to the n-frame, $\boldsymbol{\Omega}(w^b) = \begin{bmatrix} 0 & -w_z^b & w_y^b \\ w_z^b & 0 & -w_x^b \\ -w_y^b & w_x^b & 0 \end{bmatrix}$, represents the anti-symmetric matrix of the angular rate vector $\boldsymbol{w}^b$. $\boldsymbol{v}^n$ and $\boldsymbol{p}^n$ denote the velocity and position vectors, respectively. $\boldsymbol{f}^b$ denotes the specific force sensed by the accelerometer, and $\boldsymbol{w}_{ie}^n$ is the rotation rate of earth expressed in the n-frame. $\boldsymbol{g}^n$ denotes the gravitational acceleration.

The gyroscope bias $\boldsymbol{\varepsilon}_w$ and accelerometer bias $\boldsymbol{\varepsilon}_a$ are both be modeled as a first-order Markov process [47]

$$\dot{\boldsymbol{\varepsilon}}_w = -\frac{1}{T_w}\boldsymbol{\varepsilon}_w + \boldsymbol{w}_w,\ \dot{\boldsymbol{\varepsilon}}_a = -\frac{1}{T_a}\boldsymbol{\varepsilon}_a + \boldsymbol{w}_a \tag{6}$$

where T_w and T_a are the correlation time, $\boldsymbol{w}_w$ and $\boldsymbol{w}_a$ denote the corresponding driving white noise.

To reduce the system linearization modeling error, the attitude, velocity, and position error states are established, which represent the difference between the true values and the estimate by the INS. Based on the error equations of the INS [48], this study incorporates accelerometer and gyroscope error models to derive the following state vector of the ESKF:

$$\boldsymbol{x} = \begin{bmatrix} \boldsymbol{\delta\varphi}^T & \boldsymbol{\delta v}^T & \boldsymbol{\delta p}^T & \boldsymbol{\varepsilon}_w{}^T & \boldsymbol{\varepsilon}_a{}^T \end{bmatrix}^T \tag{7}$$

where $\boldsymbol{\delta\varphi}$, $\boldsymbol{\delta v}$, and $\boldsymbol{\delta p}$ represent the attitude, velocity, and position error vectors, respectively. Additionally, the system state equation is represented in matrix form, and the state-space equation is denoted as

$$\dot{\boldsymbol{x}} = \boldsymbol{f}\boldsymbol{x} + \boldsymbol{d}\boldsymbol{w} \tag{8}$$

where

$$\boldsymbol{f} = \begin{bmatrix} -\boldsymbol{w}_{ie}^n\times & \boldsymbol{0}_{3\times3} & \boldsymbol{0}_{3\times3} & -\boldsymbol{C}_b^n & \boldsymbol{0}_{3\times3} \\ (\boldsymbol{C}_b^n\boldsymbol{f}^b)\times & -2\boldsymbol{w}_{ie}^n\times & \boldsymbol{0}_{3\times3} & \boldsymbol{0}_{3\times3} & \boldsymbol{C}_b^n \\ \boldsymbol{0}_{3\times3} & \boldsymbol{I}_3 & \boldsymbol{0}_{3\times3} & \boldsymbol{0}_{3\times3} & \boldsymbol{0}_{3\times3} \\ \boldsymbol{0}_{3\times3} & \boldsymbol{0}_{3\times3} & \boldsymbol{0}_{3\times3} & -\frac{1}{T_a}\boldsymbol{I}_3 & \boldsymbol{0}_{3\times3} \\ \boldsymbol{0}_{3\times3} & \boldsymbol{0}_{3\times3} & \boldsymbol{0}_{3\times3} & \boldsymbol{0}_{3\times3} & -\frac{1}{T_g}\boldsymbol{I}_3 \end{bmatrix}$$

$$\boldsymbol{d} = \begin{bmatrix} -\boldsymbol{C}_b^n & \boldsymbol{0}_{3\times3} & \boldsymbol{0}_{3\times3} & \boldsymbol{0}_{3\times3} \\ \boldsymbol{0}_{3\times3} & \boldsymbol{C}_b^n & \boldsymbol{0}_{3\times3} & \boldsymbol{0}_{3\times3} \\ \boldsymbol{0}_{3\times3} & \boldsymbol{0}_{3\times3} & \boldsymbol{0}_{3\times3} & \boldsymbol{0}_{3\times3} \\ \boldsymbol{0}_{3\times3} & \boldsymbol{0}_{3\times3} & \boldsymbol{I}_3 & \boldsymbol{0}_{3\times3} \\ \boldsymbol{0}_{3\times3} & \boldsymbol{0}_{3\times3} & \boldsymbol{0}_{3\times3} & \boldsymbol{I}_3 \end{bmatrix}$$

$$\boldsymbol{w} = \begin{bmatrix} \boldsymbol{w}_g{}^T & \boldsymbol{w}_f{}^T & \boldsymbol{w}_w{}^T & \boldsymbol{w}_a{}^T \end{bmatrix}^T \tag{9}$$

with $\boldsymbol{I}_n$ represents the n-dimensional identity matrix, while $\boldsymbol{w}_g$ and $\boldsymbol{w}_f$ denote the measurement white noises from the gyroscopes and accelerometer, respectively.

When the OZP is identified at time k, the velocity estimate $\boldsymbol{v}_k^{ins}$ derived from INS is regarded as the observed velocity

Algorithm 1 EOGI-SLAM Algorithm

$u_t = f(\hat{p}^y_{ozp,t}, \hat{p}^y_{ozp,t-1})$ // get control vector from odometry
for each particle $m = 1$ to n **do**
  $s_t^{[m]} = s_{t-1}^{[m]} + u_t^*[m]$ // pose prediction
  for each $\theta_i \in \Theta$ **do**
    $\tilde{N}_{\theta_i,t}^{[m]} = Bres(u_t^{[m]})$
    $l_t^{[m]}(\theta_i) = \log\left(\frac{p(z_t|\theta_i,s_t)}{1-p(z_t|\theta_i,s_t)}\right) + l_{t-1}(\theta_i)$
    $l_t^{[m]}(\theta_i) = \tilde{N}_{\theta_i,t}^{[m]} \cdot l_{t-1}^{[m]}(\theta_i)$
    $p\left(\theta_i^{[m]} \mid s_{1:t}^{[m]}, z_{1:t}^{[m]}\right) = 1 - \frac{1}{1+e^{l_t^{[m]}(\theta_i)}}$ // map update
  end for
  $w_t^{[m]} = \frac{p(s_{1:t}^{[m]}|z_{1:t}^{[m]}, w_{1:t}^{[m]})}{p(s_{1:t}^{[m]}|z_{1:t-1}^{[m]}, w_{1:t}^{[m]})}$ // particle weight update
end for
$w_t = \eta \cdot w_t$ // particle weight normalization
if $D(w_t) > N_{\text{th}}$ **then**
  perform resampling
end if

error. The velocity observation equation is established as follows:

$$z_k = \delta v_k = v_k^{ins} = h_k x_k + n_{v,k} \tag{10}$$

where $h = [\mathbf{0}_{3\times 3}\ I_3\ \mathbf{0}_{3\times 9}]$, n_v is measurement noise, primarily arising because the touch point of the robot is ideally modeled as a fixed point, inevitably introducing observation errors. The error states include navigation state error and IMU bias. Navigation state errors are subtracted from the system estimate to enhance performance, while IMU bias should be removed to mitigate odometry performance degradation before the next INS update.

B. EOGI-SLAM

EOGI-SLAM fundamentally involves the joint posterior probability estimation of the robot's pose and map. According to RBPF theorem [45], (1) can be decomposed into the robot pose estimation and map estimation given the robot's pose

$$\begin{aligned} &p(\mathbf{s}_{1:t}, \Theta \mid \mathbf{z}_{1:t}, \mathbf{u}_{1:t}) \\ &\quad = p(\mathbf{s}_{1:t} \mid \mathbf{z}_{1:t}, \mathbf{u}_{1:t}) \times \prod_{i=1}^{N_\theta} p(\theta_i \mid \mathbf{s}_{1:t}, \mathbf{z}_{1:t}) \end{aligned} \tag{11}$$

where θ_i is the ith grid cell in Θ, N_θ is the whole number of grid cell.

EOGI-SLAM estimates the posterior distribution of the robot's pose and the grid map in the particle filter framework, the algorithmic overview of the EOGI-SLAM framework is shown in Algorithm 1.

$$\tilde{N}_{\theta_i,t}^{[m]} = Bres\left(p_{t-1}^{x,[m]}, p_{t-1}^{y,[m]}, p_t^{x,[m]}, p_t^{y,[m]}\right). \tag{12}$$

The algorithm first obtains the proposal distribution for the particles and then approximates the true posterior distribution by assigning different weights to each particle. Each particle consists of a pose sequence and an associated grid map, and the mth particle is represented as follows:

$$\zeta_t^{[m]} = \left\langle \mathbf{s}_{1:t}^{[m]}, \Theta^{[m]} \right\rangle \tag{13}$$

where $\mathbf{s}_{1:t}^{[m]}$ represents the pose sequence, given by $\mathbf{s}_{1:t} = \{\mathbf{s}_1, \mathbf{s}_2, \ldots, \mathbf{s}_t\}$. Here, $\mathbf{s}_t = [p_t^x, p_t^y, \psi_t]^T$, (p_t^x, p_t^y) represents the robot position, and ψ_t denotes the heading angle. Θ consists of grid cells, each maintaining its accessibility probability.

As the robot moves, the inertial odometry generates control vectors. The set of control vectors is represented as $\mathbf{u}_{1:t} = \{\mathbf{u}_1, \mathbf{u}_2, \ldots, \mathbf{u}_t\}$. Here $\mathbf{u}_t = [l_t, \Delta\psi_t]$, l_t and $\Delta\psi$ indicate the step length and heading change, respectively. The definitions given as follows:

$$l_t = \sqrt{\left(\hat{p}^x_{ozp,t} - \hat{p}^x_{ozp,t-1}\right)^2 + \left(\hat{p}^y_{ozp,t} - \hat{p}^y_{ozp,t-1}\right)^2} \tag{14}$$

$$\Delta\psi_t = \arctan\left(\frac{\hat{p}^y_{ozp,t} - \hat{p}^y_{ozp,t-1}}{\hat{p}^x_{ozp,t} - \hat{p}^x_{ozp,t-1}}\right) - \psi_{t-1} \tag{15}$$

where $\hat{p}_{ozp,t-1}$ and $\hat{p}_{ozp,t}$ are positions of adjacent OZPs estimated by the odometry.

The proposal distribution is pose prediction based on the current pose and control vector. Specifically, EOGI-SLAM uses the dead reckoning model [49] and Monte Carlo sampling [50] to predict the particle's pose

$$\begin{bmatrix} p_t^{x,[m]} \\ p_t^{y,[m]} \\ \psi_t^{[m]} \end{bmatrix} = \begin{bmatrix} p_{t-1}^{x,[m]} \\ p_{t-1}^{y,[m]} \\ \psi_{t-1}^{[m]} \end{bmatrix} + \begin{bmatrix} l_t^{*[m]} \cos\left(\psi_{t-1}^{[m]}\right) \\ l_t^{*[m]} \sin\left(\psi_{t-1}^{[m]}\right) \\ \Delta\psi_t^{*[m]} \end{bmatrix} \tag{16}$$

where

$$\begin{aligned} l_t^* &\sim \mathcal{N}\left(l_t, \sigma^l\right) \\ \Delta\psi_t^* &\sim \mathcal{N}\left(\Delta\psi_t, \sigma^{\Delta\psi}\right) \end{aligned} \tag{17}$$

with $\mathcal{N}$ denotes the Gauess distribution, σ^l and $\sigma^{\Delta\psi}$ present the corresponding standard deviations.

Considering that the accessibility probability of grid cells is affected by the traversal degree of particles, the proposed EOGI-SLAM divided each grid cell into $5 * 5$ equally sized subgrid cells, and the Bresenham algorithm [51] is employed to calculate the number of subgrid cells crossed by each motion vector

$$\tilde{N}_{\theta_i,t}^{[m]} = Bres\left(p_{t-1}^{x,[m]}, p_{t-1}^{y,[m]}, p_t^{x,[m]}, p_t^{y,[m]}\right) \tag{18}$$

where $\tilde{N}_{\theta_i,t}^{[m]}$ denotes the number of subgrid cells crossed by the motion vector in the cell θ_i at moment t, *Bres* denotes the Bresenham line algorithm.

According to Bayesian theory, the posterior accessibility probability of the grid cell θ_i can be estimated by introducing the log ratio of the inverse measurement model [41], which is denoted as

$$l_t(\theta_i) = \log \frac{p(\theta_i \mid s_{1:t}, \mathbf{z}_{1:t})}{1 - p(\theta_i \mid s_{1:t}, \mathbf{z}_{1:t})}. \tag{19}$$

The log ratio of the inverse measurement model is updated as follows:

$$l_t(\theta_i) = l_{t-1}(\theta_i) + \log \frac{p(\theta_i \mid s_t, z_t)}{1 - p(\theta_i \mid s_t, z_t)} - \log \frac{p(\theta_i)}{1 - p(\theta_i)} \tag{20}$$

$$= l_{t-1}(\theta_i) + \log \frac{p(\theta_i \mid s_t, z_t)}{1 - p(\theta_i \mid s_t, z_t)} \tag{21}$$

$$= \tilde{N}_{\theta_i,1:t} \cdot \log \frac{p(\theta_i \mid s_t, z_t)}{1 - p(\theta_i \mid s_t, z_t)} \tag{22}$$

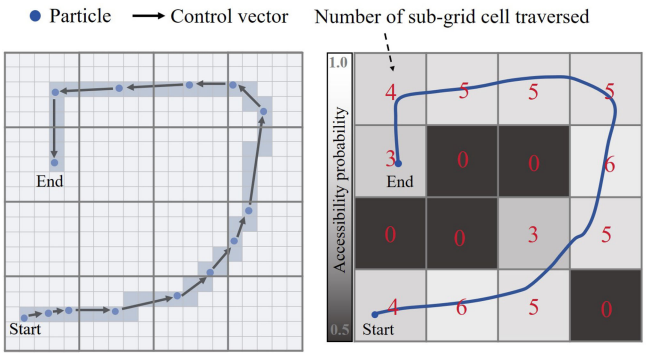


Fig. 6. Map update process of proposed EOGI-SLAM.

where $p(\theta_i)$ is the prior accessibility probability of the grid cell. Since the initial accessibility of the grid is unknown, it is set to 0.5. $p(\theta_i \mid s_t, z_t) \in (0.5, 1)$ represents the inverse observation model, which describes how motion vectors affect the accessibility probability of the grid cell. Since EOGI-SLAM uses the number of subgrids crossed by motion vectors to determine the update for the grid cell, $p(\theta_i \mid s_t, z_t)$ is set to 0.6 in this article. Then, the posterior accessibility estimate of grid cells is denoted as

$$p(\theta_i \mid s_{1:t}, z_{1:t}) = 1 - \frac{1}{1 + e^{l_t(\theta_i)}}. \tag{23}$$

The motion vector-driven map update process is shown in Fig. 6. With small-displacement motion vector inputs, the grid map is not subjected to repeated incorrect updates. Additionally, the continuity and integrity of the map are maintained with small displacement inputs.

In the particle weight update part, each particle is assigned a weight based on the difference between the proposal and target distribution. Specifically, the weight of particles is defined as the ratio of the target distribution to the proposed distribution. Thus, for the mth particle, the weight is given by

$$w_t^{[m]} = \frac{p\left(\mathbf{s}_{1:t}^{[m]}, \Theta \mid \mathbf{z}_{1:t}^{[m]}, \mathbf{u}_{1:t}^{[m]}\right)}{p\left(\mathbf{s}_{1:t}^{[m]}, \Theta \mid \mathbf{z}_{1:t-1}^{[m]}, \mathbf{u}_{1:t}^{[m]}\right)}. \tag{24}$$

According to Fast-SLAM [44] and Bayesian rule [35], the weight of the mth particle can be calculated as follows:

$$w_t^{[m]} = \eta \cdot p\left(\mathbf{z}_t^{[m]} \mid \mathbf{s}_{1:t}^{[m]}, \Theta, \mathbf{z}_{1:t-1}^{[m]}, \mathbf{u}_{1:t}^{[m]}\right) \tag{25}$$

where η is a normalization factor that ensures the sum of all particle weights equals 1. $p(\mathbf{z}_t^{[m]} \mid \mathbf{s}_{1:t}^{[m]}, \Theta, \mathbf{z}_{1:t-1}^{[m]}, \mathbf{u}_{1:t}^{[m]})$ is the accessibility probability of the grid cell where the particle is located.

After calculating the particle weights, the appropriate resampling is also required to avoid particle degradation. However, the frequent resampling process leads to the loss of particle diversity, which makes it difficult to construct effective observations. Thus, the dispersion of the particles is evaluated by calculating the variance of the particle weights, the resampling is performed when the variance exceeds the set threshold value.

Fig. 7. Unitree B1 legged robot platform and real scenarios of Test1 and Test2.

TABLE I
PARAMETERS OF XSENS MTW

Unit	Accelerometer	Gyroscope
Dimensions	3 axes	3 axes
Dynamic range	±2000 $^\circ/s$	±160 m/s^2
Bias stability	10 $^\circ/h$	0.03 mg

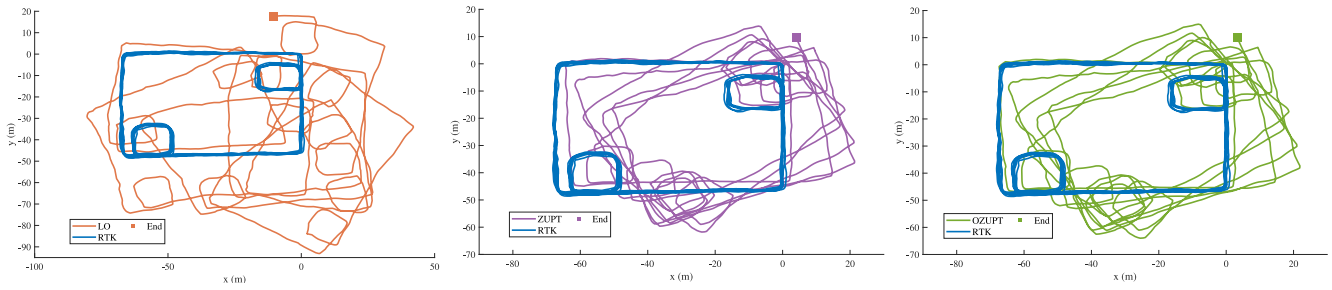
IV. EXPERIMENT AND RESULTS

In this section, the field experiments were designed and conducted to validate the effectiveness of the proposed method. The positioning performance of the proposed odometry method and the EOGI-SLAM framework was quantitatively analyzed to demonstrate their effectiveness in improving localization accuracy.

A. Experimental Description

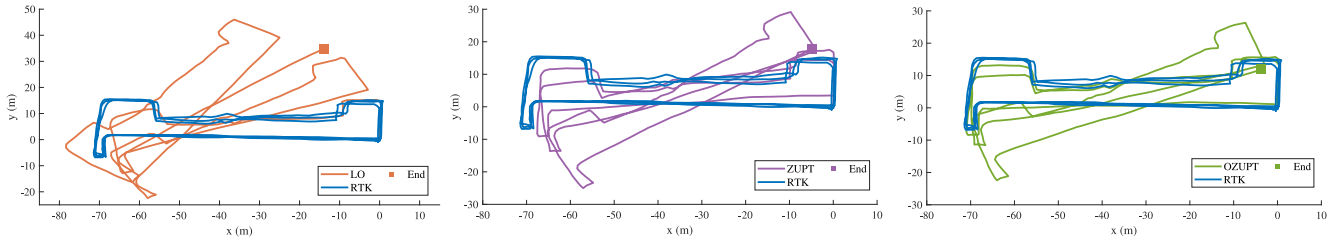
The experiments were conducted using the Unitree B1 legged robot platform, a single IMU is mounted on the robot foot to construct the proposed inertial odometry. The IMU used is the Xsens MTw, which integrates a three-axis accelerometer, gyroscope, and magnetometer, all with an output frequency of 100 Hz. The specific parameters of the IMU are listed in Table I. A real-time kinematic (RTK) positioning device was employed to obtain the ground truth of position. Two experiments are referred to as Test1 and Test2, respectively. Test1 features a 2061.5 m trajectory around a soccer field, representative of IoT scenarios like warehouse inspection. To ensure sufficient runtime, only onboard sensors and a foot-mounted IMU were used, enabling evaluation of long-term positioning performance. Test2 was conducted in a complex outdoor environment specifically designed to evaluate the robustness of the proposed algorithm under challenging surface conditions. The site included a variety of uneven terrains, such as loose gravel patches and irregular ground elevations, simulating the unstructured surfaces often encountered in real-world deployments. The robot traversed a total 569.4 m path and returned to the starting point.

In both tests, the robot's movements were remotely controlled, covering various motion modes, such as standstill, velocity changes, and in-place stepping. The starting position was set to the origin (0, 0) for consistency in analysis. The tests were conducted on a university campus with frequent pedestrian activity, naturally introducing dynamic elements, such as nearby human movement. A test operator accompanied the robot during the experiments to ensure safety and assist



(a) (b) (c)

Fig. 8. Position estimation results with three different odometry methods in Test1. (a) LO. (b) ZUPT. (c) OZUPT.



(a) (b) (c)

Fig. 9. Position estimation results with three different odometry methods in Test2. (a) LO. (b) ZUPT. (c) OZUPT.

with data collection, while also contributing to the dynamic elements in the environment. The robot platform and test scenarios are illustrated in Fig. 7.

For the validation of the proposed odometry, the traditional LO and the traditional ZUPT method are selected as the control group.

1) *Leg Odometry (LO):* The forward kinematics is applied to compute the homogeneous transformation matrix from the foot to the body based on the robot model and joint angles. The displacement and velocity changes of the robot body are then estimated recursively, determining the state of the entire robot [13].
2) *ZUPT:* The zero-velocity observation is performed during the whole stance phase to correct the cumulative localization error [25].

In the SLAM validation, Fast-LIO [29] and ORB-SLAM3 [28] are included in Test2 as baselines, representing more competitive multisensor SLAM approaches beyond the scope of inertial-only odometry. For inertial SLAM, the proposed EOGI-SLAM is evaluated against the baseline OGI-SLAM to assess its performance. In total, eight methods are considered in the comparison.

1) *Fast-LIO:* A widely used LiDAR-inertial odometry system that employs an iterated Kalman filter for tightly coupled state estimation. In our experiments, it is configured with a Hesai PandarXT-32 LiDAR operating at 10 Hz.
2) *ORB-SLAM3:* A state-of-the-art visual-inertial SLAM system that combines feature-based stereo tracking with IMU preintegration. In our tests, stereo image pairs are captured at 30 Hz.
3) *LO-OGI, ZUPT-OGI, and OZUPT-OGI:* The OGI-SLAM method serves as the back-end optimization, while the front-end odometry inputs are provided by LO, ZUPT, and OZUPT, respectively.
4) *LO-EOGI, ZUPT-EOGI, and OZUPT-EOGI:* The proposed EOGI-SLAM method acts as the back-end optimization, with front-end odometry inputs also provided by LO, ZUPT, and OZUPT, respectively.

It is worth noting that both OGI-SLAM and EOGI-SLAM rely solely on IMU data, without incorporating LiDAR, cameras, or any other external sensors.

B. Experimental Results and Analysis

1) Odometry Analysis: The position estimation results of the legged robot using different odometry methods in two tests are shown in Figs. 8 and 9, respectively. The results show that as the robot's motion time increases, the trajectories deviate from the true value under different methods. The deviation occurs because all three odometry methods rely on recursive state estimation without absolute position or heading observations. Notably, the trajectory estimated by LO shows increasing deviation due to robot modeling errors, joint encoder noise, and potential foot slippages. In contrast, both the traditional ZUPT and the proposed OZUPT methods apply velocity observations during each stance phase, providing periodic corrections to accumulated system errors.

The traditional ZUPT method assumes zero velocity across the entire stance phase, neglecting minor velocity and displacement changes due to small foot rotations. This assumption leads to an underestimation of the estimated displacement. In the OZUPT method, a single OZP is chosen in each stance phase for zero-velocity measurement, reducing the accumulation of errors from inaccurate observations. The detection results for OZPs and the velocity estimation for

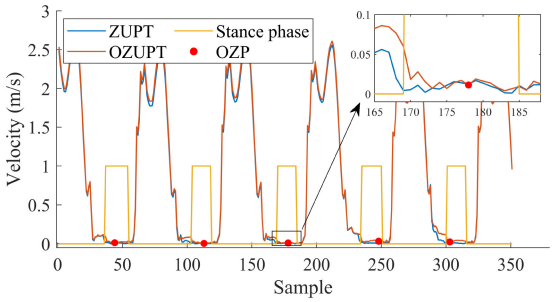


Fig. 10. Detection results of OZPs and velocity estimation results using the ZUPT and OZUPT methods.

TABLE II
POSITIONING ERRORS OF DIFFERENT ODOMETRY METHODS IN TWO TESTS

Test	Method	RMSE (m)	MAX (m)	CEP75 (m)
Test1	LO	36.13	93.44	55.53
	ZUPT	17.01	44.14	25.71
	OZUPT	**16.25**	**43.52**	**24.72**
Test2	LO	14.42	46.81	19.85
	ZUPT	7.31	23.47	10.34
	OZUPT	**5.52**	**18.67**	**7.95**

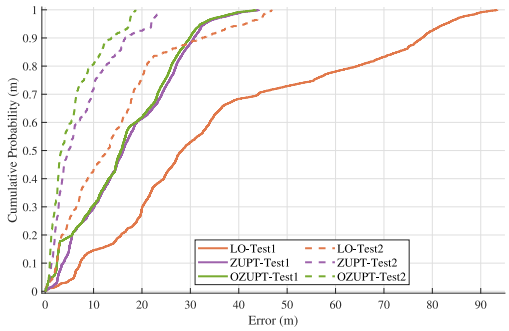


Fig. 11. CDF of positioning errors with different odometry methods in two tests.

both the ZUPT and OZUPT methods are shown in Fig. 10. During the stance phase, the velocity estimated using the ZUPT method is generally closer to zero, failing to account for the small rotational movements of the foot. As a result, it underestimates the actual velocity.

The positioning accuracy of the different methods is quantitatively analyzed by calculating the positioning error, which is defined as follows:

$$\varepsilon_k = \sqrt{\left(r^x_{k,est} - r^x_{k,rtk}\right)^2 + \left(r^y_{k,est} - r^y_{k,rtk}\right)^2} \tag{26}$$

where k is the index number of the coordinate point. $(r^x_{k,est}, r^y_{k,est})$ denotes the position estimate, and $(r^x_{k,rtk}, r^y_{k,rtk})$ is the results.

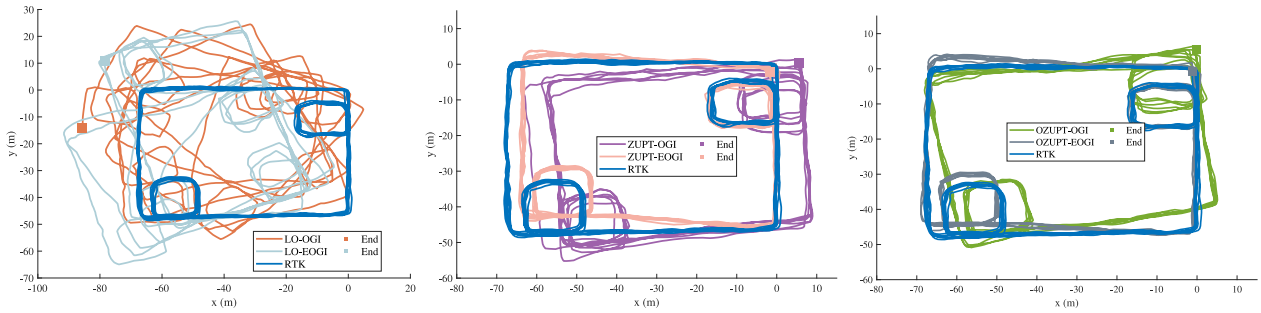
Fig. 11 presents the cumulative distribution function (CDF) of positioning errors for the two tests. The error statistics, including root mean square error (RMSE), maximum error, and circular error probable at 75% (CEP75), are summarized in Table II. The results indicate that the LO method accumulates substantial positioning errors, with RMSE values of 36.13 and 14.42 m, respectively. While the ZUPT algorithm improves accuracy, velocity observation errors still lead to RMSE values of 17.01 and 7.31 m. In contrast, the proposed OZUPT method achieves more precise velocity estimation, reducing the RMSE to 16.25 and 5.52 m.

The terrain in Test2 is more complex compared to Test1, this introduces additional challenges for maintaining accurate positioning for odometry. For two tests, the RMSE percentages of the LO method are 1.75% and 2.53%, respectively. For the ZUPT algorithm, the percentages are 0.83% and 1.28%, while the proposed method achieves 0.79% and 0.97%. These results indicate that although increased terrain complexity can degrade odometry accuracy, the proposed method maintains better performance than the baselines.

2) SLAM Analysis: In this section, the performance of different SLAM methods is validated and analyzed. The following default algorithm configuration is used for inertial SLAM methods: the number of particles is set to 2000, and the grid length is 1.0 m.

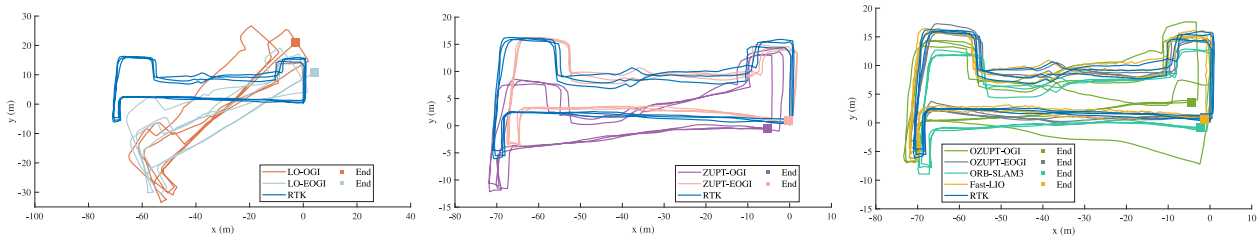
The position estimation results of SLAM methods with different odometry inputs are presented in Figs. 12 and 13. Due to the low positioning accuracy of LO, both the LO-OGI and LO-EOGI methods face challenges in achieving effective loop closure. In some cases, these methods produce localized false loopbacks, which distort the trajectory and compromise the accuracy of the positioning system. In contrast, the ZUPT-OGI and OZUPT-OGI methods demonstrate superior positioning performance, maintaining overall heading stability and generally achieving better loop closure outcomes. However, the imbalance in accessibility probabilities across the grid map in the OGI-SLAM algorithm occasionally leads to incorrect local loop closures. In contrast, the ZUPT-EOGI and OZUPT-EOGI methods successfully mitigate such errors through improved map update strategies. On the other hand, while the ZUPT-EOGI method effectively corrects heading errors, velocity estimation bias results in a smaller overall displacement than the ground truth, leading to noticeable localization errors.

The partial grid map generated by the OZUPT-OGI method is illustrated in Fig. 14, where different colors represent varying accessibility probabilities. A notable discrepancy in accessibility probabilities is observed in areas without loop closures. Specifically, due to differences in step length inputs, the grid cells in Region2 are updated more frequently than those in Region1, leading to a significant disparity in accessibility probabilities. Fig. 15 depicts the erroneous loop closure process in the OZUPT-OGI method. During the pose prediction phase, a particle driven by Monte Carlo sampling [50] moves into a high-accessibility probability region, increasing its weight and ultimately causing a false loop closure. The imbalance in grid cell accessibility probabilities heightens the risk of incorrect loop closures, negatively impacting the overall accuracy and performance of the SLAM algorithm. In contrast, the partial grid map generated by the OZUPT-EOGI method, shown in Fig. 16, exhibits minimal differences in accessibility probabilities between Region1



(a) (b) (c)

Fig. 12. Position estimation results using different SLAM methods in Test1. (a) LO as the front-end input. (b) ZUPT as the front-end input. (c) OZUPT as the front-end input.



(a) (b) (c)

Fig. 13. Position estimation results using different SLAM methods in Test2. (a) LO as the front-end input. (b) ZUPT as the front-end input. (c) OZUPT as the front-end input, with comparisons, including ORB-SLAM3 and Fast-LIO.

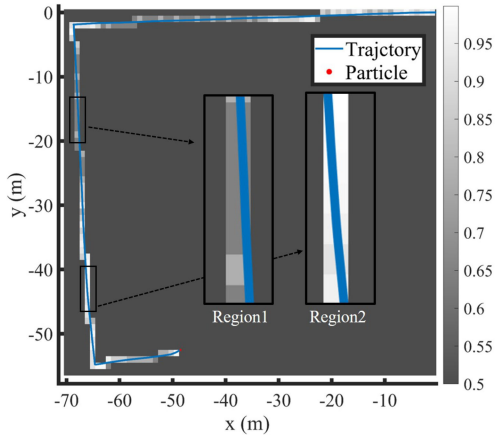
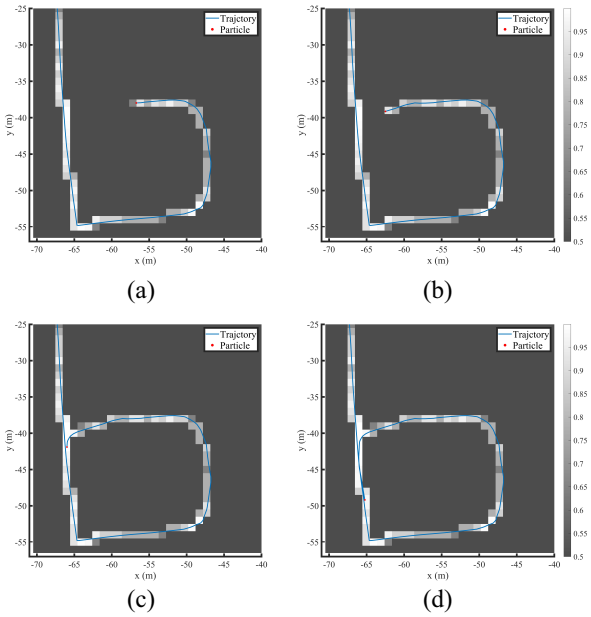


Fig. 14. Partial grid maps estimated by the OZUPT-OGI method in Test1.



(a) (b) (c) (d)

Fig. 15. Process of the wrong loop closure of particle using the OZUPT-OGI method in Test1 due to imbalance of accessibility probability.

and Region2, mitigating the likelihood of erroneous loop closures.

The CDFs of positioning errors for different SLAM methods in two tests are shown in Figs. 17 and 18, respectively, with the corresponding error statistics summarized in Table III. Compared to other OGI- and EOGI-based variants, our proposed OZUPT-EOGI method achieves superior performance. In Test1, OZUPT-EOGI achieves an RMSE of 1.97 m, significantly lower than ZUPT-OGI at 9.29 m, ZUPT-EOGI at 3.09 m, and OZUPT-OGI at 5.59 m. In Test2, which involves more complex and challenging terrain, OZUPT-EOGI remains robust with an RMSE of 1.76 m, outperforming ZUPT-OGI at 5.67 m, ZUPT-EOGI at 2.65 m, and OZUPT-OGI at 3.50 m. These results demonstrate that our method outperforms across different testing scenarios.

In addition to the inertial SLAM baselines, Test2 includes benchmark comparisons with two representative multisensor SLAM systems: ORB-SLAM3 and Fast-LIO. Among

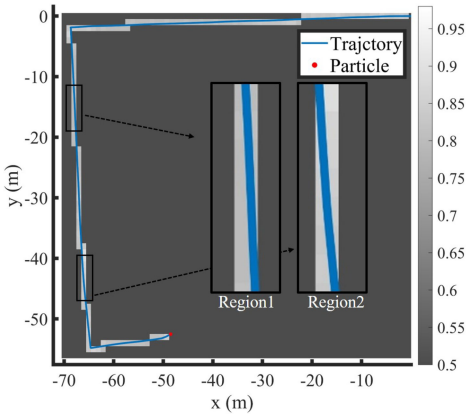


Fig. 16. Partial grid map estimated by OZUPT-EOGI method in Test1.

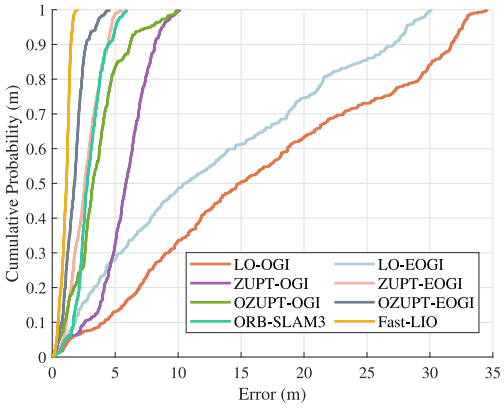


Fig. 18. CDF of positioning errors with different SLAM methods in Test2.

TABLE III
POSITIONING ERRORS OF DIFFERENT SLAM METHODS

Test	Method	RMSE (m)	MAX (m)	CEP75 (m)
Test1	LO-OGI	37.30	96.35	58.87
	LO-EOGI	33.44	80.69	47.19
	ZUPT-OGI	9.29	14.9	11.04
	ZUPT-EOGI	3.09	7.32	4.95
	OZUPT-OGI	5.59	9.94	7.21
	OZUPT-EOGI	1.97	5.34	3.11
Test2	LO-OGI	16.60	34.54	26.32
	LO-EOGI	12.58	30.11	20.32
	ZUPT-OGI	5.67	10.14	7.05
	ZUPT-EOGI	2.65	5.55	3.54
	OZUPT-OGI	3.50	9.96	4.43
	OZUPT-EOGI	1.76	4.54	2.22
	ORB-SLAM3	2.93	5.91	3.62
	Fast-LIO	1.05	2.03	1.33

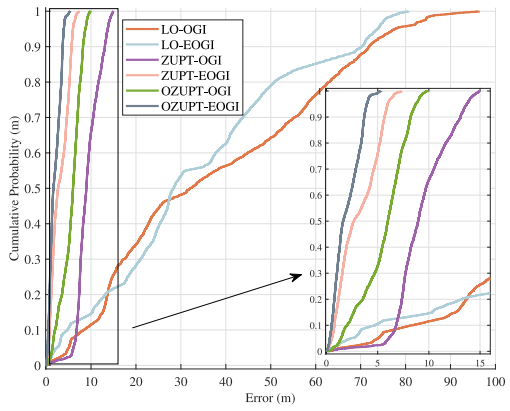


Fig. 17. CDF of positioning errors with different SLAM methods in Test1.

all evaluated methods, Fast-LIO achieves the best overall performance, with an RMSE of 1.05 m, and also outperforms other methods in terms of maximum error and CEP75. However, Fast-LIO depends on both IMU and LiDAR inputs, which significantly increases system cost. In contrast, the proposed OZUPT-EOGI method, relying solely on an IMU, achieves a competitive RMSE of 1.76 m. Notably, it also outperforms ORB-SLAM3, which yields an RMSE of 2.93 m. The reduced accuracy of ORB-SLAM3 could be attributed to dynamic environments and image jitter caused by legged robot motion. These results underscore the robustness and efficiency of our method, highlighting its potential as a cost-effective alternative to more complex multisensor SLAM systems.

To further analyze drift accumulation, we introduce the cumulative drift (cDrift) metric to quantify how positioning errors accumulate over distance. Conceptually equivalent to the cumulative absolute trajectory error (c-ATE) used in [52], the cDrift is computed by summing instantaneous positioning errors at 1-s intervals and normalizing them with respect to the travelled distance. This approach accounts for differences in output frequency across methods and provides a fair evaluation of drift accumulation.

For clarity, LO-OGI and LO-EOGI are excluded due to their comparatively poor performance, and the results of the two tests are shown in Figs. 19 and 20, respectively. In Test1, where only inertial methods are evaluated, the proposed OZUPT-EOGI method achieves the best performance, exhibiting the lowest cDrift among all compared methods. In Test2, FAST-LIO exhibits the lowest cDrift, primarily due to the inherent advantages of LiDAR in providing accurate and dense geometric information. Meanwhile, ORB-SLAM3 exhibits a certain degree of drift growth. Notably, our proposed OZUPT-EOGI method relies solely on a single IMU and is inherently unaffected by environmental features. It ranks second overall, outperforming other inertial SLAM methods and showing strong potential for reliable localization.

To evaluate the computational efficiency and resource demands of the proposed method, we compared the average runtime and memory usage of four SLAM systems in Test2: 1) FAST-LIO; 2) ORB-SLAM3; 3) OGI-SLAM; and 4) EOGI-SLAM. The results are summarized in Table IV. All methods were tested under the same hardware configuration: Intel Core i5-10400F CPU with 16 GB of RAM. The sensor data rates differ across methods: ORB-SLAM3 uses a camera operating at 30 Hz, FAST-LIO processes LiDAR scans at 10 Hz, and both OGI-SLAM and EOGI-SLAM use IMU data sampled at 100 Hz. Therefore, we define the average runtime in this article as the time required to process one second of input data, expressed in milliseconds per second (ms/s), and use it as a unified metric for computational complexity.

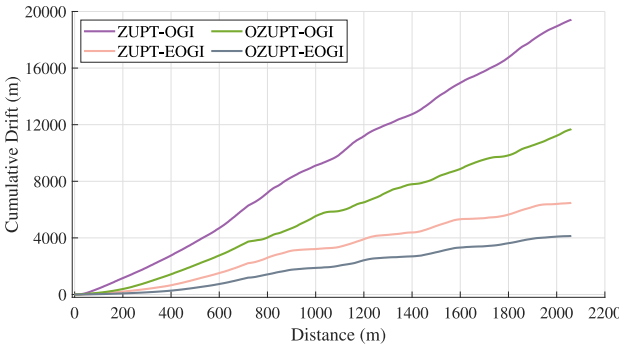


Fig. 19. cDrift with different SLAM methods in Test1.

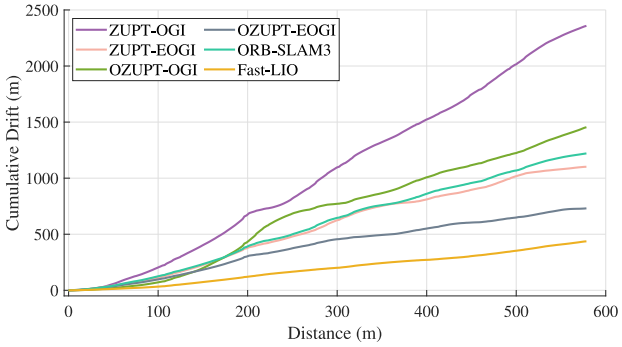


Fig. 20. cDrift with different SLAM methods in Test2.

TABLE IV
COMPARISON OF RUNTIME EFFICIENCY AND MEMORY USAGE

Methods	Sensor Rate (Hz)	Avg. Runtime (ms/s)	Avg. Memory usage (MB)
FAST-LIO	10	171.3	493.1
ORB-SLAM3	30	885.9	2269.7
OGI-SLAM	100	82.6	50.3
EOGI-SLAM	100	96.7	64.2

As observed in the results, although EOGI-SLAM introduces additional processing compared to the baseline OGI-SLAM due to the overhead of map updating, its average runtime increases by only 14.1%, rising from 82.6 to 96.7 ms/s. In contrast, it achieves a substantial improvement in positioning accuracy, with the RMSE reduced from 3.50 to 1.76 m, representing a 49.7% improvement. Compared with FAST-LIO and ORB-SLAM3, EOGI-SLAM demonstrates significantly lower runtime and memory usage, mainly due to the reduced computational burden of processing only IMU data rather than large-scale LiDAR or visual inputs. These results indicate that the proposed method maintains competitive performance and offers a more efficient and lightweight alternative, making it well-suited for deployment on resource-constrained platforms. Furthermore, based on the observed runtime of 96.7 ms/s, the EOGI-SLAM method can comfortably support real-time execution at the IMU sampling rate of 100 Hz and exhibits sufficient computational headroom to operate at even higher input rates if necessary.

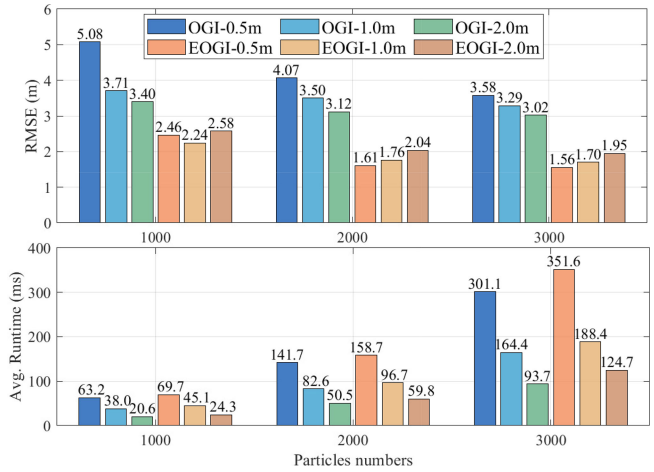


Fig. 21. RMSE and average runtime of OZUPT-OGI and OZUPT-EOGI methods under varying parameters in Test2. For instance, "OGI-0.5 m" represents the OZUPT-OGI method with a grid length of 0.5 m.

To comprehensively evaluate the performance of the proposed EOGI-SLAM system under different parameter configurations, the RMSE and average runtime of OZUPT-OGI and OZUPT-EOGI across varying particle numbers and grid resolutions are compared. The corresponding results are presented in Fig. 21. As the number of particles increases and the grid size decreases, the average runtime tends to increase due to higher computational demands, while the improvement in localization accuracy becomes less significant. These observations indicate that further increasing the particle count or refining the grid resolution yields limited gains in accuracy and leads to inefficient use of computational resources. Selecting a suitable configuration, such as the default experimental setting used in this article with 2000 particles and a 1-m grid, achieves a balanced tradeoff between computational cost and localization performance. Furthermore, the algorithm parameters can be flexibly adjusted based on the performance constraints and resource availability of different platforms, allowing the system to be effectively deployed across a wide range of hardware environments.

V. DISCUSSION

A. Potential Bottlenecks

Although the proposed method improves position estimation for legged robots, certain limitations remain. A key limitation lies in the complete reliance on a single IMU for localization, inertial odometry inherently suffers from cumulative errors caused by sensor noise and bias in the absence of absolute reference observations. The drift in our system is primarily due to the accumulation of IMU bias and sensor noise.

Moreover, several motion-related factors can further compromise localization performance. During the stance phase, small vibrations in the legs may cause the IMU to deviate from the ideal zero-velocity assumption, introducing minor but persistent velocity estimation errors. Moreover, foot slippage can hinder the accurate detection of stance phases and OZPs, leading to the absence of effective velocity correction and further contributing to drift. These issues tend to introduce

noticeable uncertainty, especially in terrains densely covered with loose gravel or characterized by extremely smooth surfaces.

Another important bottleneck stems from algorithmic parameters, specifically the number of particles and the grid resolution. Increasing the number of particles can enhance the robustness and granularity of the posterior pose distribution, potentially improving localization accuracy. Similarly, a finer grid resolution yields more detailed maps and more precise likelihood evaluations. However, both adjustments significantly increase computational cost and memory consumption. However, there is often a bottleneck in the increase in accuracy, while the running time continues to increase. Inefficient parameter selection may result in unnecessary resource consumption without corresponding performance benefits. This challenge is especially critical for deployment on platforms with limited computational resources.

B. Trade-Offs With Multisensor SLAM

Multisensor SLAM approaches, such as ORB-SLAM3 and FAST-LIO, combine IMU measurements with visual or geometric information to effectively constrain drift and improve localization accuracy. These methods benefit from exteroceptive sensing, which facilitates global consistency and high precision. However, they often rely on high-quality sensor data and demand precise calibration and substantial computational resources. Such requirements can pose challenges for robotic platforms with limited power, payload, or sensing capabilities.

In contrast, the proposed EOGI-SLAM system operates solely with a single IMU, without reliance on external environmental perception. This IMU-only configuration offers several key advantages, including reduced system complexity, lower hardware cost, and reliable operation in featureless environments, such as low-light conditions and geometrically sparse scenes. Despite these benefits, the IMU-only design also has inherent limitations, the system becomes more susceptible to drift over long trajectories.

In summary, the choice between IMU-only and multisensor SLAM systems represents a tradeoff between system simplicity and long-term localization accuracy. EOGI-SLAM offers a lightweight, computationally efficient, and cost-effective alternative, particularly well-suited for environments where visual or LiDAR data are unreliable or unavailable. For long-range missions or structurally rich environments, multisensor SLAM methods may still provide superior global performance.

C. Future Work

Future work will focus on improving the system's robustness and long-term accuracy by integrating additional sensors, such as vision, LiDAR, or UWB. Although the current system operates in an IMU-only configuration, it is inherently modular in design. The algorithm decouples IMU integration, zero-velocity detection, and pose optimization, and is capable of outputting both raw integrated motion estimates and optimized poses. This modular structure provides a natural interface for incorporating additional exteroceptive sensors, such as vision, LiDAR, or UWB.

Another promising direction is the extension of EOGI-SLAM to multirobot collaborative scenarios. By sharing occupancy grid maps or local pose estimates, a swarm of lightweight agents, such as IoT devices could perform distributed SLAM in a scalable and communication-efficient manner. In addition, future experiments will explore more diverse and challenging environments, including uneven terrains, such as gravel paths and outdoor slopes, to further evaluate the system's adaptability to real-world deployment conditions. We recognize the importance of assessing robustness in multifloor indoor environments. To this end, future work will focus on enhancing the control framework of the legged robot for safe operation on stairs and inclined surfaces, and on integrating more precise ground-truth acquisition systems to enable reliable evaluation in RTK-degraded indoor settings.

Finally, considering the resource constraints typical of low-power embedded systems, future work will also explore strategies for improving computational and energy efficiency. For instance, adaptive resampling or dynamic adjustment of particle counts can help reduce computational load while maintaining localization accuracy. These strategies aim to improve the practicality of EOGI-SLAM in large-scale, long-duration, and energy-constrained robotic applications.

VI. Conclusion

This article introduces an inertial odometry and EOGI-SLAM for legged robots. The inertial odometry involves recognizing each OZP within the stance phase using IMU signals, and the OZUPT is performed at OZP in the ESKF framework to correct odometry drift. The proposed inertial odometry is straightforward to implement, operating as a complete and independent state estimator without relying on the kinematic modeling of the robot. Additionally, the EOGI-SLAM system is introduced, and the motion vector-driven map update is proposed to address map discretization and accessibility probability imbalances, thereby reducing the likelihood of incorrect particle updates. The system is comprehensively validated on the Unitree B1 robot, experimental results demonstrate that the proposed odometry and EOGI-SLAM system significantly improve position estimation accuracy.

References

[1] W. Shao, H. Luo, F. Zhao, H. Tian, S. Yan, and A. Crivello, "Accurate indoor positioning using temporal–spatial constraints based on Wi-Fi fine time measurements," *IEEE Internet Things J.*, vol. 7, no. 11, pp. 11006–11019, Nov. 2020.

[2] 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.

[3] Y. Lizarribar, D. Giustiniano, G. Bovet, and V. Lenders, "SkyPos: Real-world evaluation of self-positioning with aircraft signals for IoT devices," *IEEE J. Sel. Areas Commun.*, vol. 42, no. 1, pp. 134–145, Jan. 2024.

[4] H. Kabir, M.-L. Tham, and Y. C. Chang, "Internet of Robotic Things for mobile robots: Concepts, technologies, challenges, applications, and future directions," *Digit. Commun. Netw.*, vol. 9, no. 6, pp. 1265–1290, 2023.

[5] J. Carpentier and P.-B. Wieber, "Recent progress in legged robots locomotion control," *Current Robot. Rep.*, vol. 2, no. 3, pp. 231–238, 2021.

[6] P. M. Wensing, M. Posa, Y. Hu, A. Escande, N. Mansard, and A. D. Prete, "Optimization-based control for dynamic legged robots," *IEEE Trans. Robot.*, vol. 40, pp. 43–63, 2024.

[7] J.-H. Kim et al., "Legged robot state estimation with dynamic contact event information," *IEEE Robot. Autom. Lett.*, vol. 6, no. 4, pp. 6733–6740, Oct. 2021.

[8] Y. Xu, Y. S. Shmaliy, S. Bi, X. Chen, and Y. Zhuang, "Extended Kalman/UFIR filters for UWB-based indoor robot localization under time-varying colored measurement noise," *IEEE Internet Things J.*, vol. 10, no. 17, pp. 15632–15641, Sep. 2023.

[9] M. Li, T. Xu, M. Guan, F. Gao, and N. Jiang, "LEO-constellation-augmented multi-GNSS real-time PPP for rapid re-convergence in harsh environments," *GPS Solut.*, vol. 26, p. 29, Jan. 2022.

[10] M. Sun et al., "Simultaneous WiFi ranging compensation and localization for indoor NLoS environments," *IEEE Commun. Lett.*, vol. 26, no. 9, pp. 2052–2056, Sep. 2022.

[11] Y. Zhuang, C. Zhang, J. Huai, Y. Li, L. Chen, and R. Chen, "Bluetooth localization technology: Principles, applications, and future trends," *IEEE Internet Things J.*, vol. 9, no. 23, pp. 23506–23524, Dec. 2022.

[12] F. Che et al., "Feature-based generalized Gaussian distribution method for NLoS detection in ultra-wideband (UWB) indoor positioning system," *IEEE Sensors J.*, vol. 22, no. 19, pp. 18726–18739, Oct. 2022.

[13] G. Roston and E. Krotkov, "Dead reckoning navigation for walking robots," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 1992, pp. 607–612.

[14] J. Hidalgo-Carrio, A. Babu, and F. Kirchner, "Static forces weighted jacobian motion models for improved Odometry," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2014, pp. 169–175.

[15] P.-C. Lin, H. Komsuoglu, and D. E. Koditschek, "A leg configuration measurement system for full-body pose estimates in a hexapod robot," *IEEE Trans. Robot.*, vol. 21, no. 3, pp. 411–422, Jun. 2005.

[16] M. Bloesch et al., "State estimation for legged robots-consistent fusion of leg kinematics and IMU," *Robotics*, vol. 17, pp. 17–24, Jul. 2013.

[17] N. Rotella, M. Bloesch, L. Righetti, and S. Schaal, "State estimation for a humanoid robot," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2014, pp. 952–958.

[18] V. Agrawal, S. Bertrand, R. Griffin, and F. Dellaert, "Proprioceptive state estimation of legged robots with kinematic chain modeling," in *Proc. IEEE-RAS 21st Int. Conf. Humanoid Robots (Humanoids)*, 2022, pp. 178–185.

[19] R. Buchanan, M. Camurri, F. Dellaert, and M. Fallon, "Learning inertial odometry for dynamic legged robot state estimation," in *Proc. Conf. Robot Learn.*, 2022, pp. 1575–1584.

[20] S. Yang, Q. Yang, R. Zhu, Z. Zhang, C. Li, and H. Liu, "State estimation of hydraulic quadruped robots using invariant-EKF and kinematics with neural networks," *Neural Comput. Appl.*, vol. 36, no. 5, pp. 2231–2244, 2024.

[21] S. Yang, Z. Zhang, Z. Fu, and Z. Manchester, "Cerberus: Low-drift visual-inertial-leg odometry for agile locomotion," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, 2023, pp. 4193–4199.

[22] D. Wisth, M. Camurri, and M. Fallon, "VILENS: Visual, inertial, LiDAR, and leg Odometry for all-terrain legged robots," *IEEE Trans. Robot.*, vol. 39, no. 1, pp. 309–326, Feb. 2023.

[23] S. Yang, H. Choset, and Z. Manchester, "Online kinematic calibration for legged robots," *IEEE Robot. Autom. Lett.*, vol. 7, no. 3, pp. 8178–8185, Jul. 2022.

[24] M. Maravgakis, D.-E. Argiropoulos, S. Piperakis, and P. Trahanias, "Probabilistic contact state estimation for legged robots using inertial information," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, 2023, pp. 12163–12169.

[25] J. Wahlström and I. Skog, "Fifteen years of progress at zero velocity: A review," *IEEE Sensors J.*, vol. 21, no. 2, pp. 1139–1151, Jan. 2021.

[26] R. Mur-Artal, J. M. M. Montiel, and J. D. Tardós, "ORB-SLAM: A versatile and accurate monocular SLAM system," *IEEE Trans. Robot.*, vol. 31, no. 5, pp. 1147–1163, Oct. 2015.

[27] R. Mur-Artal and J. D. Tardós, "ORB-SLAM2: An open-source SLAM system for monocular, stereo, and RGB-D cameras," *IEEE Trans. Robot.*, vol. 33, no. 5, pp. 1255–1262, Oct. 2017.

[28] C. Campos, R. Elvira, J. J. G. Rodríguez, J. M. Montiel, and J. D. Tardós, "ORB-SLAM3: An accurate open-source library for visual, visual–inertial, and multimap SLAM," *IEEE Trans. Robot.*, vol. 37, no. 6, pp. 1874–1890, Dec. 2021.

[29] 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.

[30] H. Pu, J. Luo, G. Wang, T. Huang, H. Liu, and J. Luo, "Visual SLAM integration with semantic segmentation and deep learning: A review," *IEEE Sensors J.*, vol. 23, no. 19, pp. 22119–22138, Oct. 2023.

[31] P. Karkus, S. Cai, and D. Hsu, "Differentiable SLAM-Net: Learning particle SLAM for visual navigation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 2815–2825.

[32] K. Liu and M. Cao, "DLC-SLAM: A robust LiDAR-SLAM system with learning-based denoising and loop closure," *IEEE/ASME Trans. Mechatronics*, vol. 28, no. 5, pp. 2876–2884, Oct. 2023.

[33] P. Robertson, M. Angermann, and B. Krach, "Simultaneous localization and mapping for pedestrians using only foot-mounted inertial sensors," in *Proc. 11th Int. Conf. Ubiquitous Comput.*, 2009, pp. 93–96.

[34] M. Angermann and P. Robertson, "FootSLAM: Pedestrian simultaneous localization and mapping without exteroceptive sensors—Hitchhiking on human perception and cognition," *Proc. IEEE*, vol. 100, pp. 1840–1848, May 2012.

[35] P. Stano, Z. Lendek, J. Braaksma, R. Babuška, C. de Keizer, and A. J. den Dekker, "Parametric Bayesian filters for nonlinear stochastic dynamical systems: A survey," *IEEE Trans. Cybern.*, vol. 43, no. 6, pp. 1607–1624, Dec. 2013.

[36] B. Wang, J. Zhu, Z. Ma, Z. Deng, and M. Fu, "Improved particle filter-based matching method with gravity sample vector for underwater gravity-aided navigation," *IEEE Trans. Ind. Electron.*, vol. 68, no. 6, pp. 5206–5216, Jun. 2021.

[37] M. G. Puyol, P. Robertson, and O. Heirich, "Complexity-reduced FootSLAM for indoor pedestrian navigation using a geographic tree-based data structure," *J. Locat. Based Services*, vol. 7, no. 3, pp. 182–208, 2013.

[38] P. Robertson, M. Angermann, and M. Khider, "Improving simultaneous localization and mapping for pedestrian navigation and automatic mapping of buildings by using online human-based feature labeling," in *Proc. IEEE/ION Posit., Locat. Navig. Symp.*, 2010, pp. 365–374.

[39] M. Garcia Puyol, D. Bobkov, P. Robertson, and T. Jost, "Pedestrian simultaneous Localization and mapping in multistory buildings using inertial sensors," *IEEE Trans. Intell. Transp. Syst.*, vol. 15, no. 4, pp. 1714–1727, Aug. 2014.

[40] M. Hardegger, D. Roggen, S. Mazilu, and G. Tröster, "ActionSLAM: Using location-related actions as landmarks in pedestrian SLAM," in *Proc. Int. Conf. Indoor Posit. Indoor Navig. (IPIN)*, 2012, pp. 1–10.

[41] Y. Ding, Z. Xiong, W. Li, Z. Cao, and Z. Wang, "Pedestrian navigation system with trinal-IMUs for drastic motions," *Sensors*, vol. 20, no. 19, p. 5570, 2020.

[42] Y. Ding, Z. Xiong, J. Xiong, Z. Cao, and W. Li, "Distributed collaborative pedestrian inertial SLAM with unknown initial relative poses," *IEEE Internet Things J.*, vol. 9, no. 21, pp. 21632–21647, Nov. 2022.

[43] Y. Ding, Z. Xiong, J. Xiong, Y. Cui, and Z. Cao, "OGI-SLAM2: A hybrid map SLAM framework grounded in inertial-based SLAM," *IEEE Trans. Instrum. Meas.*, vol. 71, pp. 1–14, Sep. 2022.

[44] T. Bailey, J. Nieto, and E. Nebot, "Consistency of the FastSLAM algorithm," in *Proc. Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, 2006, pp. 424–429.

[45] G. Grisetti, C. Stachniss, and W. Burgard, "Improving grid-based SLAM with Rao-blackwellized particle filters by adaptive proposals and selective Resampling," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2005, pp. 2432–2437.

[46] H. Li et al., "Adaptive threshold-based ZUPT for single IMU-enabled wearable pedestrian localization," *IEEE Internet Things J.*, vol. 10, no. 13, pp. 11749–11760, Jul. 2023.

[47] D. Unsal and K. Demirbas, "Estimation of deterministic and stochastic IMU error parameters," in *Proc. IEEE/ION Posit., Locat. Navig. Symp.*, 2012, pp. 862–868.

[48] Q. Zhang, X. Niu, and C. Shi, "Impact assessment of various IMU error sources on the relative accuracy of the GNSS/INS systems," *IEEE Sensors J.*, vol. 20, no. 9, pp. 5026–5038, May 2020.

[49] Y. Wu, J. Kuang, and X. Niu, "Wheel-INS2: Multiple MEMS IMU-based dead reckoning system with different configurations for wheeled robots," *IEEE Trans. Intell. Transp. Syst.*, vol. 24, no. 3, pp. 3064–3077, Mar. 2023.

[50] D. Luengo, L. Martino, M. Bugallo, V. Elvira, and S. Särkkä, "A survey of monte carlo methods for parameter estimation," *EURASIP J. Adv. Signal Process.*, vol. 2020, pp. 1–62, May 2020, doi: 10.1186/s13634-020-00675-6.

[51] F. L. Gaol, "Bresenham algorithm: Implementation and analysis in raster shape," *J. Comput.*, vol. 8, no. 1, pp. 69–78, 2013.

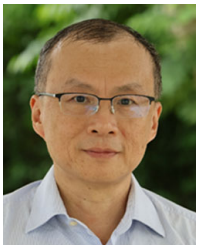
[52] M. Brossard and S. Bonnabel, "Learning wheel odometry and IMU errors for Localization," in *Proc. Int. Conf. Robot. Autom. (ICRA)*, 2019, pp. 291–297.

Xiaodong Li received the B.S. degree in automation from the Nanjing University of Aeronautics and Astronautics, Nanjing, China, in 2020, where he is currently pursuing the Ph.D. degree with the College of Automation Engineering.

He is currently a visiting Ph.D. student with the School of Electrical and Electronic Engineering, Nanyang Technological University, Singapore. His research interests include state estimation, indoor localization, and multisource fusion.

Zhi Xiong received the M.S. and Ph.D. degrees from Nanjing University of Aeronautics and Astronautics (NUAA), Nanjing, China, in 2001 and 2004, respectively.

He joined NUAA, where he has been a Full Professor with the College of Automation Engineering, since 2011. In 2013, he was an Academic Visiting Fellow with the University of Southern California, Los Angeles, CA, USA. He has ten years of experience in the inertial navigation field and has led more than 30 navigation system development projects. His main research areas include inertial navigation, Small aircraft navigation, brain-like navigation, and multisource fusion.



Lipo Wang (Senior Member, IEEE) received the B.Sc. degree from National University of Defense Technology, Changsha, China, in 1983, and the Ph.D. degree from Louisiana State University, Baton Rouge, LA, USA, in 1988.

He is presently with the School of Electrical and Electronic Engineering, Nanyang Technological University, Singapore. He has 400+ publications, two patents and 14 000+ Google Scholar citations. He was keynote speaker for 40+ international conferences. His research interest is artificial intelligence for image and data processing.

Dr. Wang is a Co-Editor-in-Chief of *International Journal of Computational Intelligence and Applications*, a Senior Editor of IEEE TRANSACTIONS ON SYSTEMS, MAN, AND CYBERNETICS: SYSTEMS, and an Associate Editor of IEEE TRANSACTIONS ON NEURAL NETWORKS AND LEARNING SYSTEMS. He is/was associate editor/editorial board member of 30+ international journals, including two other IEEE Transactions, and guest editor for 10+ journal special issues. He was a member of the Board of Governors of the International Neural Network Society, IEEE Computational Intelligence Society, and the IEEE Biometrics Council. He served as a CIS Vice President for Technical Activities and a Chair of Emergent Technologies Technical Committee, as well as a Chair of Education Committee of the IEEE Engineering in Medicine and Biology Society. He was President of the Asia–Pacific Neural Network Assembly (APNNA, renamed as APNNS – "Society") and received the APNNA Excellent Service Award. He was Founding Chair of both the EMBS Singapore Chapter and CIS Singapore Chapter.

Yan Cui received the B.Sc. degree from the Nanjing University of Aeronautics and Astronautics, Nanjing, China, in 2021, where he is currently pursuing the Ph.D. degree with the Navigation Research Center.

His main fields of research interest include indoor inertial navigation and indoor multisource navigation.