Eng. Sci., 2025, **38**, 1883

Engineered Science

DOI: https://dx.doi.org/10.30919/es1883

LILO: Light Detection and Ranging, Inertial and Leg Odometry Combined Simultaneous Localization and Mapping Based on Kalman Filter for Legged Robots

Guangrong Chen,[1,2,3,*] Qizhe Yang,[1] Mengqi Yang,[1] Guangxin Zhang,[1] Mengqiu Mo[1] and Yuxiang Lin[4]

Abstract

The Simultaneous Localization and Mapping (SLAM) performance of a mobile robot is affected by many factors, including the algorithm, sensor scheme and mobile platform, etc. Compared with wheeled robots, the SLAM performance of legged robots is weak with the same algorithm and sensor configurations since the unstable legged locomotion leads to the body vibration, which is unfavorable to the data sampling of sensors. To address this issue, we propose an improved LiDAR-Inertial-Leg Odometry (LILO) algorithm that tightly couples light detection and ranging (LiDAR), inertial measurement unit (IMU), and leg odometry pre-integration factors within a factor graph framework. The algorithm leverages IMU and leg odometry information for drift-free motion priors while LiDAR provides structural constraints, and all measurements are fused through Kalman filtering and nonlinear optimization in Georgia Tech Smoothing and Mapping (GTSAM). The Laser SLAM and visual SLAM with different algorithm are compared on wheeled and legged robots, and comparative experiments demonstrate that the proposed LILO framework effectively mitigates the impact of vibrational noise, enabling a legged robot to achieve reliable SLAM with an overall mapping accuracy of 4.6% mean relative absolute error in challenging environments, significantly outperforming standard single-sensor and leg-agnostic fusion approaches.

Keywords: Simultaneous localization and mapping (SLAM); IMU correction; Multi sensors; Legged robots; Kalman filter.
Received: 19 July 2025; Revised: 28 October 2025; Accepted: 12 November 2025
Article type: Research article.

1. Introduction

How mobile robots can better perceive their environment in complex scenarios is a crucial question because it determines the quality of the robot's subsequent positioning and navigation functions.[1] In recent years, the emerging multi-sensor information fusion technology has effectively addressed this issue. It has contributed a comprehensive solution for the intelligent movement of robots, and this approach is continuously expanding and extending.[2]

With the complexity of robot application scenarios, relying solely on a single sensor is far from providing sufficient useful information for robots. Therefore, experts and scholars in the field of SLAM have gradually conducted in-depth research on multi-sensor fusion technology,[3,4] By integrating different types of information collected, robots can also enhance their perception of the surrounding environment.[5]

Early LiDAR-IMU odometry systems, such as LOAM,[6] demonstrated the feasibility of real-time mapping through scan- to-map registration, while later works like tightly-coupled lidar inertial odometry via smoothing and mapping (LIO-SAM)[7] and Fast-LIO/Fast-LIO2[8,9] introduced tightly coupled smoothing or iterated Kalman filtering to achieve higher accuracy and robustness. Ye *et al*.[10] proposed a tightly coupled LiDAR-IMU odometry and mapping approach that improved drift resistance in complex environments. More recently, Wisth *et al*.[11] extended this line of research to include leg odometry, showing that fusing proprioceptive leg data with LiDAR and IMU can significantly benefit all-terrain robots.

For quadrupedal platforms, vibration and intermittent foot-ground contact make the above methods less reliable. The perception of robot body motion through kinematic legs provides a complementary source of velocity information that, when properly fused, can suppress high-frequency noise while compensating low-frequency drift. Miki *et al*.[12] explored how

---

[1] *Robotics Research Center, Beijing Jiaotong University, Beijing, 100044, China*

[2] *Key Laboratory of Vehicle Advanced Manufacturing, Measuring and Control Technology, Ministry of Education, Beijing Jiaotong University, Beijing, 100044, China*

robust locomotion policies can benefit from rich exteroceptive perception, highlighting the need for accurate state estimation under natural disturbances. Chen and Hong[13] further analyzed LiDAR-vision perception systems for quadruped robots, confirming the importance of multi-modal sensing in unstructured scenes.

Currently, commonly used sensor fusion methods include Bayesian estimation, Kalman filtering, fuzzy reasoning, etc..[14] Li Jing *et al* investigated the open street map-based autonomous navigation for the four wheel-legged robot via three dimensional LiDAR (3D-LiDAR) and Charge-Coupled Device (CCD) camera.[15] Zhu Zhenkun *et al* proposed an effective training strategy in reference to the operating mode of SLAM systems, so that the feature extraction network can better adapt to the SLAM system for pose estimation and mapping.[16] Shen Shaojie *et al* proposed a monocular visual-inertial SLAM solution named VINS- Mono.[17] Roumeliotis *et al* improved the algorithm's efficiency by marginalizing the state vector of the Extended Kalman Filter (EKF).[18] Anastasios *et al* proposed a Multi-State Constrained Visual Inertial Navigation Algorithm (MSCCKF).[19] Carlos Campos *et al* presented a oriented fast and brief SLAM3 (ORB-SLAM3), the first system able to perform visual, visual-inertial and multimap SLAM with monocular, stereo and RGB-D cameras, using pin-hole and fisheye lens models.[4] Leutengger *et al* predicted the state by using IMU dynamic equations.[20] Forster *et al* introduced a manifold-based IMU discrete pre- integration algorithm.[21] Alan *et al* proposed a Time Delay Estimation (TDE) algorithm similar to time synchronization.[22] Rigatos *et al* presented an algorithm that uses encoder and ultrasonic data as the state vector for estimation.[23] Li Shengyu *et al* proposed two-step LiDAR/camera/IMU spatial and temporal calibration based on continuous-time trajectory estimation. This proposed method is target-free and has no assumptions about the sensor configuration.[24] Li Haisong *et al* proposed an accurate, robust, and real-time LiDAR-inertial SLAM method for solid-state LiDARs in degenerated environments.[25] Xu Lei *et al* proposed an EPLF-VINS: real-time monocular visual-inertial SLAM with efficient point-line flow features.[26] Chen Jiaying *et al* presented a computationally efficient LiDAR-inertial odometry method that directly parameterizes ground vehicle poses on special Euclidean group in 2 dimensions (SE(2)).[27] Zhang Yi *et al* proposed a tightly coupled LiDAR+ IMU fusion system with loop optimization to address the data drifting problem.[28] Cao Like *et al* using the foundation framework of VINS-Mono, implemented the fusion of IMU and RGB-D cameras.[29] Gong Xuerui *et al* achieved the fusion of vision and inertial navigation under the ORB-SLAM2 system.[30] They used the EKF algorithm to fuse the pose output from ORB-SLAM2 with IMU data to improve localization accuracy.

However, most of these frameworks were designed for wheeled robots with stable locomotion. When applied to legged robots, the inherent vibrations and discontinuous foot-terrain interactions introduce significant high-frequency noise and drift, especially for LiDAR odometry and kinematic leg odometry. Recent progress such as VILENS[11] and the Fast-LIO family[8,9] demonstrates that careful probabilistic fusion of IMU, LiDAR, and proprioceptive signals can alleviate these challenges. To tackle this gap, we propose LILO, a tightly coupled SLAM framework that fuses LiDAR, IMU, and leg odometry via a Kalman filter. Building upon insights from LOAM,[6] LIO-SAM,[7] Fast-LIO [8], Fast-LIO2 [9], and proprioceptive-aided approaches[11-13] our method aims to deliver accurate and robust state estimation for legged robots in vibration-prone and uneven environments.

Aiming to improve the SLAM performance for legged robots, in this paper, IMU is combined with LiDAR and leg odometry to address the vibration problem in legged locomotion.

2. System platform and model

2.1 Platform of quadruped robot

The hardware configuration for the proposed body-environment cooperative perception task consists of a LiDAR sensor, an RGB-D depth camera, and an IMU. These components are mounted on a lightweight quadruped platform to provide complementary proprioceptive and exteroceptive data for state estimation and mapping.

The experimental platform employed in this study is the Go1 quadruped robot manufactured by Unitree Technology Co., Ltd., as illustrated in Fig. 1. Go1 is a compact, electrically actuated robot designed for agile locomotion over uneven terrain, which makes it an appropriate carrier for evaluating SLAM algorithms under vibration-prone conditions.

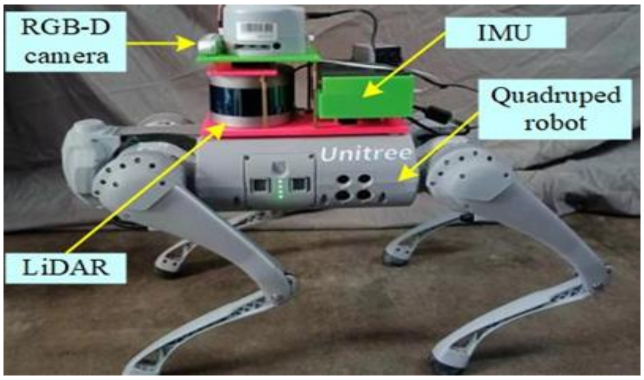


Fig. 1: Experimental quadruped platform (Unitree Go1) equipped with a 16-line RoboSense LiDAR, an Orbbec Gemini2 RGB-D camera, and a Witmotion IMU.

For body perception, a ten-axis IMU module

---

[3]Tangshan Research Institute of Beijing Jiaotong University, Tangshan, 063000, China
[4]Dundee International Institute, Central South University, Changsha, 410004, China
*Email: grchen@bjtu.edu.cn (Guangrong Chen)

manufactured by Witmotion Technology Co., Ltd. is integrated on the main body of the robot. This module provides triaxial angular velocity, linear acceleration, and orientation measurements, enabling real-time estimation of the robot's attitude and dynamic motion state. Accurate inertial data are essential for compensating drift in LiDAR odometry and for stabilizing pose estimation during rapid gait transitions.

For environmental perception, a RoboSense 16-line mechanical LiDAR is mounted on the top of the platform to acquire high-resolution 3D point clouds of the surroundings. The LiDAR delivers a balance between range, vertical resolution, and update frequency, which is suitable for outdoor and semi-structured environments. To further enrich scene geometry and provide dense visual cues, an Orbbec Gemini2 RGB-D camera is installed alongside the LiDAR. The camera outputs aligned color and depth images, supplying additional point cloud information that can support spatial localization, obstacle recognition, and semantic mapping.

The overall sensory arrangement allows the robot to simultaneously capture its own motion state and detailed environmental features, forming the foundation for the proposed LiDAR-Inertial-Leg odometry (LILO) framework.

2.2 Kinematics of quadruped robot

Leg kinematics provides a fundamental basis for estimating the velocity of a quadruped robot, under the assumption that there is no slippage between the feet and the terrain and that the supporting surface remains rigid. By establishing an appropriate coordinate system and using the Denavit-Hartenberg (DH) convention, the spatial relationship between the robot body and the end-effector (foot) of each leg can be systematically derived.

Fig. 2 illustrates the coordinate frames adopted in this work, taking the right-front leg as an example. The origin of the body coordinate frame *{B}* is defined at the geometric center of the robot base and is used as the reference point for SLAM state estimation. The parameters *l, w,* and h_B denote the body length, body width, and body height, respectively. Each leg is modelled as a serial three-joint chain: a hip joint rotating about the vertical axis, a thigh joint, and a shank joint. Their spatial configuration is described using the DH parameters listed in Table 1, where θ_i is the joint angle, d_i the link offset, a_i the link length, and α_i the link twist.

Using the DH representation, the homogeneous transformation matrices between successive coordinate frames are given in Eq. (1).

$$
\begin{aligned}
\mathrm{BT}_0&=\begin{bmatrix}0&0&1&l/2\\0&1&0&-w/2\\-1&0&0&0\\0&0&0&1\end{bmatrix},\\
0\mathrm{T}_1&=\begin{bmatrix}c_1&0&s_1&0\\s_1&0&-c_1&0\\0&1&0&l_0\\0&0&0&1\end{bmatrix},\\
1\mathrm{T}_2&=\begin{bmatrix}c_2&-s_2&0&l_2c_2\\s_2&c_2&0&l_2s_2\\0&0&1&l_1\\0&0&0&1\end{bmatrix},\\
2\mathrm{T}_3&=\begin{bmatrix}c_3&-s_3&0&l_3c_3\\s_3&c_3&0&l_3s_3\\0&0&1&0\\0&0&0&1\end{bmatrix}.
\end{aligned}
\tag{1}
$$

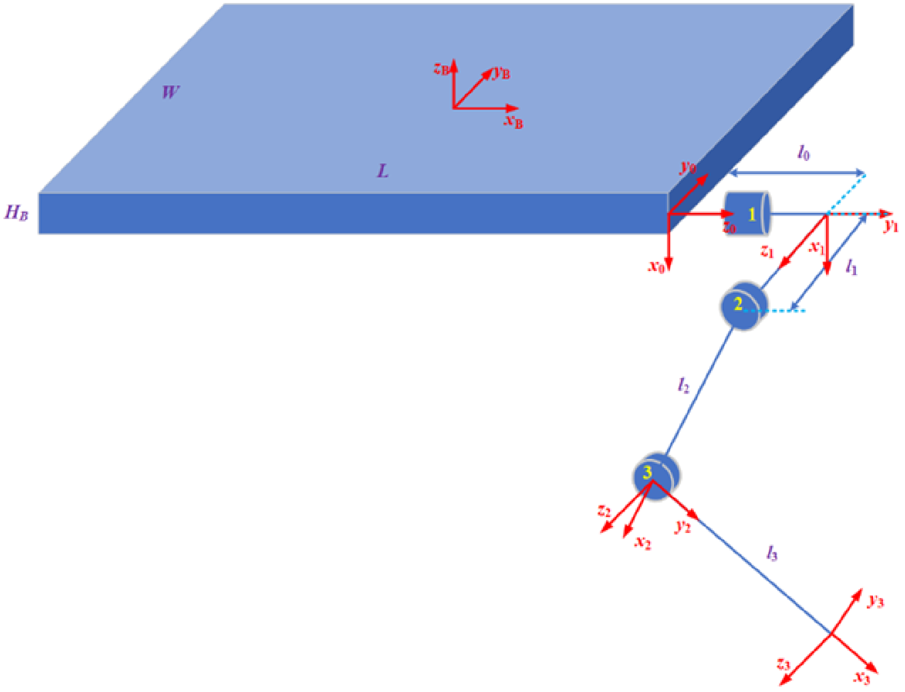


Fig. 2: Coordinate frames of the quadruped robot (right-front leg as an example).

Table 1: Denavit-Hartenberg parameters of the right-front leg.

Joint	θ_i	d_i	a_i	α_i
i=1	θ_1	l_0	0	$\pi/2$
i=2	θ_2	l_1	l_2	0
i=3	θ_3	0	l_3	0

Multiplying the above matrices yields the forward kinematics (Eq. (2)) from the body frame *{B}* to the foot frame *{3}*.

$$BT_3 = {}^{B}T_0{}^{0}T_1{}^{1}T_2{}^{2}T_3 = \begin{bmatrix} s_{23} & c_{23} & 0 & P_x \\ s_1c_{23} & -s_1s_{23} & -c_1 & P_y \\ -c_1c_{23} & c_1s_{23} & -s_1 & P_z \\ 0 & 0 & 0 & 1 \end{bmatrix}, \tag{2}$$

where $s_{23} = sin(\theta_2 + \theta_3)$ and $c_{23} = cos(\theta_2 + \theta_3)$. The position of the end-effector in frame *{B}* is obtained as shown in Eq. (3).

$$\begin{cases} P_x = \dfrac{L}{2} + l_0 + l_2s_2 + l_3s_{23}, \\ P_y = -\dfrac{W}{2} - l_1c_1 + l_2s_1c_2 + l_3s_1c_2c_3 - l_3s_1s_2s_3, \\ P_z = -l_1s_1 - l_2c_1c_2 - l_3c_1c_2c_3 + l_3c_1s_2s_3. \end{cases} \tag{3}$$

To relate joint velocities to the Cartesian velocity of the foot, the analytical Jacobian matrix J (Eq. (4)) is derived by differentiating the position vector with respect to the joint variables:

$$J = \begin{bmatrix} J_{11} & J_{12} & J_{13} \\ J_{21} & J_{22} & J_{23} \\ J_{31} & J_{32} & J_{33} \end{bmatrix}, \tag{4}$$

where

$$\begin{aligned} &J_{11} = 0, \quad J_{12} = l_2c_2 + l_3c_{23}, \quad J_{13} = l_3c_{23}, \\ &J_{21} = l_1s_1 + l_2c_1c_2 + l_3c_1c_2c_3 - l_3c_1s_2s_3, \\ &J_{22} = -l_2s_1s_2 - l_3s_1c_2s_3 - l_3s_1s_2c_3, \\ &J_{23} = -l_3s_1c_2s_3 - l_3s_1s_2c_3, \\ &J_{31} = -l_1c_1 + l_2s_1c_2 + l_3s_1c_2c_3 - l_3s_1s_2s_3, \\ &J_{32} = l_2c_1s_2 + l_3c_1s_2c_3 + l_3c_1c_2s_3, \\ &J_{33} = l_3c_1c_2s_3 + l_3c_1s_2c_3. \end{aligned} \tag{5}$$

The Jacobian establishes a linear mapping between the joint velocity vector and the Cartesian velocity of the foot tip. This relationship (Eq. (5)) will be exploited in the subsequent section to estimate the velocity of the robot's base point from measured joint motions.

3. Multi-sensor fusion SLAM

The proposed LILO framework adopts a hierarchical fusion strategy tailored to the unique challenges of SLAM on legged robots. Unlike wheeled platforms, legged locomotion introduces high-frequency vibrations and discontinuous foot-ground contacts, which degrade the accuracy of single-sensor or loosely coupled solutions. To address this issue, we integrate complementary information sources in a tightly coupled manner.

At the core of our system are three key factors: the IMU pre-integration factor, the LiDAR odometry factor, and the leg odometry pre-integration factor. The IMU pre-integration provides high-rate motion priors that enable de-skewing of LiDAR scans and improve real-time responsiveness, but it is prone to drift over long trajectories. LiDAR odometry contributes precise geometric constraints from point cloud registration, yet its performance can deteriorate under strong body vibrations or rapid leg movements, resulting in scan misalignment. The leg odometry, derived from joint kinematics, offers a drift-free estimate of the robot's velocity during stance phases, but it becomes unreliable when feet slip or leave the ground.

A Kalman filter serves as the central fusion hub, dynamically weighting these heterogeneous measurements. When the feet are firmly in contact with the terrain, the estimator prioritizes leg odometry for its stability. During swing phases or when slippage is detected, the filter smoothly shifts reliance to the IMU and LiDAR cues. This adaptive mechanism allows the system to exploit the strengths of each modality while compensating for their weaknesses, thereby mitigating vibration-induced degradation that often hampers traditional LiDAR-inertial or vision-inertial SLAM on quadruped robots.

3.1 Multi - sensor fusion

Multi-sensor fusion in SLAM aims to combine the complementary characteristics of different perception modalities, yielding state estimates that are more accurate and robust than those from any single source. For mobile robots, precise state estimation and dense mapping are critical for reliable navigation and task execution.

Vision-based SLAM methods estimate motion by tracking features across consecutive images. While computationally efficient, these approaches are sensitive to illumination and texture, and they often accumulate drift in long-term or large-scale operations. LiDAR-based methods, by contrast, are largely invariant to lighting changes and can directly acquire rich 3D geometric data. Recent advances in long-range, high-resolution 3D LiDAR have further increased their suitability for mapping complex environments.

Among LiDAR-inertial fusion algorithms, several representative methods have been developed. LOAM (LiDAR Odometry and Mapping)[31] pioneered real-time scan-to-map registration and used IMU data only for point cloud de-skewing, without incorporating inertial measurements into the optimization process. LeGO-LOAM[32] proposed a lightweight solution optimized for ground vehicles, which constructs high-quality maps quickly but depends heavily on ground segmentation and lacks loop- closure capabilities. Livox-LOAM[33] targeted solid-state LiDAR with narrow fields of view, but its robustness in cluttered environments is limited.

Considering these characteristics, the present study adopts LeGO-LOAM as the baseline owing to its efficiency and accuracy in structured scenes. We extend it by introducing IMU pre-integration factors, enabling tighter coupling between LiDAR and inertial measurements, and by incorporating loop-closure detection to suppress long-term drift. These enhancements, combined with the proposed leg odometry factor, yield a multi-layer fusion SLAM system capable of maintaining high precision even under the dynamic and vibration-rich conditions encountered by quadruped robots.

3.2 Fusion system architecture

LeGO-LOAM is a representative lightweight LiDAR odometry and mapping algorithm; however, it operates as an open-loop system. While its efficiency and suitability for structured scenes make it attractive, its absence of loop-closure detection often leads to cumulative drift during prolonged operation or when the robot traverses rough terrain. Misregistration of feature points can accumulate over time, eventually resulting in noticeable map distortion.

To address these limitations, we build upon the LeGO-LOAM framework and introduce several enhancements. First, an IMU+Leg pre-integration module is added to the front end, supplying high-frequency motion priors for de-skewing raw LiDAR scans and compensating for motion distortion in point clouds. This module jointly integrates IMU measurements and leg-odometry increments, while a Kalman filter fuses their outputs to generate a refined velocity and pose prior, effectively suppressing IMU drift and filtering out vibration noise in leg odometry. Second, a loop-closure module is integrated to suppress long-term drift by detecting revisited locations and adding corresponding constraints. Finally, the back end employs a factor graph optimizer that fuses LiDAR odometry, inertial/leg information, and loop-closure constraints to estimate a globally consistent trajectory. The overall architecture of the improved system is shown in Fig. 3.

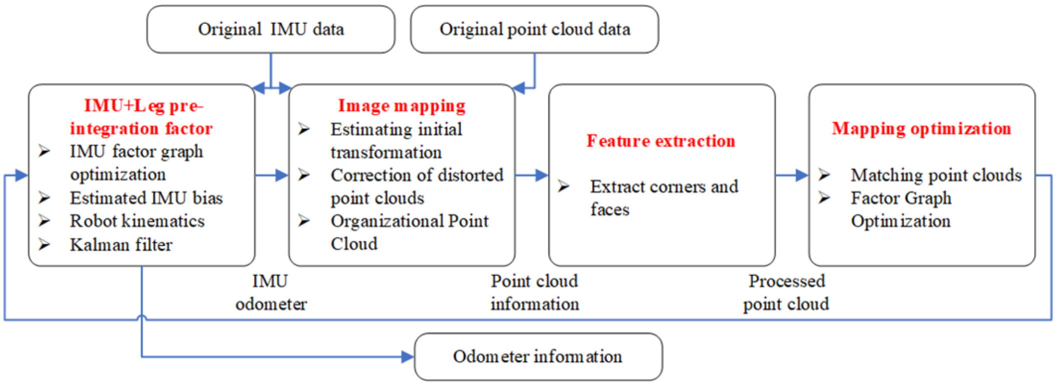


Fig. 3: Architecture of the improved LiDAR SLAM algorithm with IMU+Leg pre-integration, Kalman filtering, and loop closure.

The algorithm adopts a tightly coupled fusion strategy, in which LiDAR, IMU, and leg odometry measurements are processed in four main stages:

(1) IMU+Leg pre-integration and Kalman filtering: Raw inertial data and joint-based leg odometry are pre-integrated to produce relative motion estimates between consecutive keyframes. A Kalman filter fuses these two sources online, mitigating IMU integration drift and filtering high-frequency disturbances from leg measurements. The resulting velocity and pose priors are published at the IMU rate to support real-time state propagation.

(2) Scan undistortion and mapping: Using the fused priors as initial guesses, raw point clouds are de-skewed and organized to correct distortions caused by platform motion. An initial transformation is obtained for each scan.

(3) Feature extraction: From the corrected point clouds, edge and planar features are extracted. These geometric primitives serve as constraints for subsequent registration and optimization.

(4) Mapping and optimization: A factor graph is constructed to jointly optimize poses, IMU+Leg bias terms, and map consistency. Scan-to-map matching refines the current pose, and loop-closure detections inject additional constraints when revisiting previously mapped regions.

As illustrated in Fig. 4, the state estimation problem is formulated within a factor graph framework. Nodes represent robot states—such as position, orientation, and velocity—while edges (factors) encode constraints derived from sensor

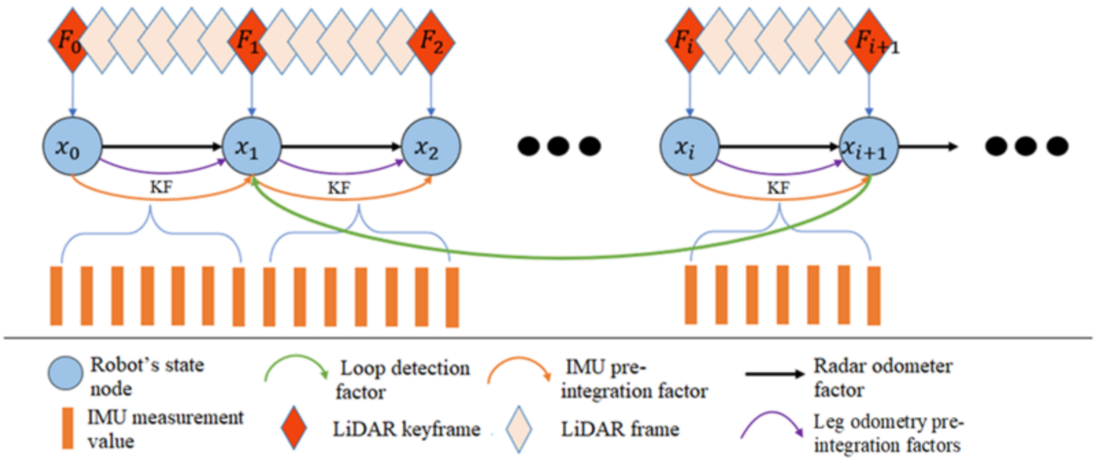


Fig. 4: Factor graph representation of the proposed multi-sensor SLAM algorithm.

observations:

- IMU+Leg pre-integration factors impose continuous motion constraints between consecutive states. Their combined information, refined by the Kalman filter, provides smooth yet drift-reduced priors for optimization.
- LiDAR odometry factors capture geometric alignment residuals between scans, offering precise but vibration-sensitive pose estimates.
- Loop-closure factors introduce strong global constraints when a previously mapped area is recognized, effectively bounding accumulated drift.

Each factor is associated with a covariance matrix that quantifies its reliability. During optimization, the solver—such as iSAM2 in the GTSAM library—balances the relative weights of all constraints to obtain the Maximum A Posteriori (MAP) estimate of the entire trajectory. This approach leverages the complementary strengths of LiDAR, IMU, and leg odometry, while systematically accounting for their uncertainties.

From a probabilistic perspective, estimating the robot's trajectory can be interpreted as solving a nonlinear least-squares problem under Gaussian noise assumptions. The robot poses act as variables to be determined, while sensor measurements define a set of constraint equations. By continuously adding new factors as data arrive, the system incrementally refines the trajectory and map, achieving high accuracy and robustness even in vibration-rich or long-duration operations.

3.3 Leg odometry pre-integration factors and kalman filter

Accurate estimation of the motion state is essential for the control and navigation of quadruped robots. Errors in state estimation directly degrade controller performance, causing unnecessary oscillations or trajectory deviations. Among all state variables, the linear velocity of the center of mass (CoM) is particularly critical, as it serves as a direct feedback signal for gait planning and stability control. However, reliable velocity acquisition is challenging: there is no dedicated sensor for measuring CoM velocity, and available measurements are often corrupted by noise or bias.

Two main approaches are commonly used to estimate the CoM velocity:

- **Kinematics-based estimation:** Under the assumptions that (i) no slippage occurs between the feet and the ground, and (ii) during flight phases the CoM velocity remains constant, the robot can be modeled as a serial manipulator with the stance foot acting as the base. By differentiating the forward kinematics of the legs, the velocity of the CoM can be computed. This method is drift-free, but the estimation accuracy depends on joint encoder precision and is susceptible to noise introduced by multiple computational stages.
- **Inertial-based estimation:** The onboard IMU provides direct measurements of the CoM linear acceleration. By compensating for sensor bias and gravitational acceleration, and then integrating over time, the CoM velocity can be obtained. While this approach yields smooth results in the short term, integration inevitably leads to drift as errors accumulate.

Each of the above methods has complementary strengths and weaknesses: kinematics-based velocity is free from long-term drift but sensitive to high-frequency disturbances and contact condition changes, whereas IMU integration produces smooth estimates but suffers from zero drift over time. To combine their advantages, we introduce a Kalman filter to fuse the outputs from both sources.

The Kalman filter is formulated as a probabilistic state estimator, recursively producing the optimal estimate of the robot's velocity in the least-squares sense. The process model uses the IMU-derived acceleration after bias compensation to propagate the velocity state forward in time. Whenever one or more feet are in stable contact with the ground, the velocity obtained from leg odometry pre-integration is treated as a measurement update, correcting for integration drift and suppressing low-frequency noise. In addition, the filter jointly estimates the IMU acceleration bias, which is one of the primary causes of long-term error in inertial integration.

The state vector is therefore defined as shown in Eq. (6).

$$\boldsymbol{x} = \begin{bmatrix} \boldsymbol{v}_{\text{CoM}} \\ \boldsymbol{b}_a \end{bmatrix}, \tag{6}$$

where $\boldsymbol{v}_{\text{CoM}} \in \mathbb{R}^3$ is the CoM linear velocity and $\boldsymbol{b}_a \in \mathbb{R}^3$ denotes the accelerometer bias. The prediction step integrates the corrected IMU acceleration, while the update step leverages the leg-derived velocity observations during stance phases.

By tightly coupling the leg odometry pre-integration factors with the IMU measurements through the Kalman filter, the proposed method achieves velocity estimates that are both drift-resistant and robust to vibration. These refined estimates form an essential part of the overall SLAM framework, providing reliable motion priors for LiDAR scan de-skewing and for improving the accuracy of factor-graph optimization.

Firstly, we clarify the estimation objective of the Kalman filter. The quantity to be estimated in this study is the robot's linear velocity (Eq. (7)) along the three body-fixed axes, namely

$$v = \begin{bmatrix} v_x \\ v_y \\ v_z \end{bmatrix}. \tag{7}$$

Secondly, we establish the state-space formulation for the filter. Since the raw acceleration measurements from the IMU are affected by a slowly varying bias (Eq. (8)), denoted by

$$\boldsymbol{b} = \begin{bmatrix} b_x \\ b_y \\ b_z \end{bmatrix}, \tag{8}$$

this bias is included in the state vector together with the velocity. The continuous-time state model expresses how the velocity evolves according to the measured acceleration (after bias correction), while the bias itself is assumed to follow a random walk:

$$\begin{bmatrix} \dot{v} \\ \dot{b} \end{bmatrix} = \begin{bmatrix} 0 & -I \\ 0 & 0 \end{bmatrix} \begin{bmatrix} v \\ b \end{bmatrix} + \begin{bmatrix} I \\ 0 \end{bmatrix} a_{\text{IMU}} + \begin{bmatrix} \eta_{\text{IMU}} \\ 0 \end{bmatrix}, \tag{9}$$

where $a_{\text{IMU}} = [a_x, a_y, a_z]^T$ is the raw accelerometer output (including bias), η_{IMU} is the accelerometer measurement noise, and $\mathbf{I}$ is a 3 × 3 identity matrix.

The observation model is derived from leg kinematics. Assuming that the stance feet do not slip and that the supporting terrain is rigid, the body velocity can be computed from the end-effector velocity of each contacting leg. Thus, the observation equation is written as

$$v_{\text{leg}} = \begin{bmatrix} I & 0 \end{bmatrix} \begin{bmatrix} v \\ b \end{bmatrix} + \eta_{\text{leg}}, \tag{10}$$

where η_{leg} denotes the uncertainty of the kinematic velocity estimation, assumed independent of η_{IMU}.

Intuitively, the process model in Eq. (9) describes the velocity propagation using the acceleration measurements corrected by the estimated bias, while the bias itself is modeled as a constant with small process noise. The observation model in Eq. (10) enforces that, when a leg is in stable contact, the velocity estimated from kinematics $\boldsymbol{v}_{\text{leg}}$ should match the true body velocity $\mathbf{v}$ up to measurement noise.

The kinematic velocity used in the measurement update (Eq. (11)) is obtained by fusing the contributions of all stance legs:

$$\boldsymbol{v}_{\text{leg}} = \begin{cases} \dfrac{\sum_{i=1}^{n} F_i \boldsymbol{v}_i}{\sum_{i=1}^{n} F_i}, & \text{stance phase}, \\ \text{last valid value}, & \text{swing phase}, \end{cases} \tag{11}$$

where n is the number of feet currently in contact with the ground, $\boldsymbol{v}_i$ is the velocity of the base point estimated from the i-th stance leg, and F_i is the corresponding contact indicator or weighting coefficient. When all legs are in the air, the filter holds the most recent valid measurement until new contact information becomes available.

By modeling velocity and accelerometer bias jointly, and by correcting the IMU integration with drift-free leg kinematics whenever contacts exist, the Kalman filter produces a smooth, low-noise estimate of the robot's center-of-mass velocity, which is essential for accurate state estimation and downstream SLAM optimization.

Let Ts denote the system sampling period. By discretizing the continuous-time state and observation equations, we obtain the following discrete-time formulation of the Kalman filter:

$$\begin{cases} X(k) = \begin{bmatrix} a & -T_s \\ 0 & a \end{bmatrix} X(k-1) + \begin{bmatrix} T_s \\ 0 \end{bmatrix} a_{\text{IMU}}(k-1) + \begin{bmatrix} T_s \eta_{\text{IMU}}(k) \\ 0 \end{bmatrix}, \\ v_{\text{leg}}(k) = \begin{bmatrix} 1 & 0 \end{bmatrix} X(k) + \eta_{\text{leg}}(k), \end{cases} \tag{12}$$

where $X(k) = [v(k)\ b(k)]^T$ is the state vector at time k, $v_{\text{leg}}(k)$ is the velocity obtained from leg kinematics, and $a_{\text{IMU}}(k-1)$ is the accelerometer measurement at the previous step.

To estimate the robot's CoM velocity and the IMU bias at step k, the covariance of process and observation noise must be specified. The process noise covariance matrix Q and the observation noise covariance R are defined as

$$Q = \begin{bmatrix} q_{\text{leg}} & 0 \\ 0 & q_{\text{IMU}} \end{bmatrix}, \quad R = r_{\text{leg}}, \tag{13}$$

where q_{leg} and q_{IMU} represent the uncertainty levels of the leg-derived velocity and IMU bias estimation, respectively. A smaller value indicates higher confidence in the corresponding quantity.

Using the state equation, a one-step prediction of the state can be written as

$$X(k|k-1) = AX(k-1|k-1) + BU(k), \tag{14}$$

with

$$A = \begin{bmatrix} 1 & -T_s s \\ 0 & 1 \end{bmatrix}, \quad B = \begin{bmatrix} T_s s \\ 0 \end{bmatrix}, \tag{15}$$

where s is a unit scaling factor (or identity for multi-dimensional cases) and $U(k)$ denotes the input $a_{\text{IMU}}(k-1)$.

The predicted covariance of the state is then given by

$$P(k|k-1) = AP(k-1|k-1)A^T + Q, \tag{16}$$

where $P(k|k-1)$ is the covariance of the predicted state.

At the update stage, the state estimate is corrected using the measurement $Z(k) = v_{\text{leg}}(k)$:

$$X(k|k) = X(k|k-1) + K(k)(Z(k) - HX(k|k-1)), \tag{17}$$

where the observation matrix is Eq. (18):

$$H = [1 \quad 0], \tag{18}$$

and the Kalman gain is computed as Eq. (19):

$$K(k) = P(k|k-1)H^T(HP(k|k-1)H^T + R)^{-1}. \tag{19}$$

Finally, the covariance of the updated state is obtained by Eq. (20):

$$P(k|k) = (I - K(k)H)P(k|k-1). \tag{20}$$

By iterating Eqs. (12)-(17), the Kalman filter provides an unbiased and minimum-variance estimate of the robot's center- of-mass velocity. After filtering, the estimated velocity is denoted by $v_{\text{KF}}(k)$. To further enhance smoothness and suppress residual high-frequency noise, a first-order low-pass filter (Eq. (21)) is applied:

$$v_{\text{KF-LF}}(k) = v_{\text{KF-LF}}(k-1) + \frac{1}{1+1/(2\pi T f_c)}(v_{\text{KF}}(k) - v_{\text{KF-LF}}(k-1)), \tag{21}$$

where T is the sampling period, which is 10 ms in both simulations and prototype experiments, and f_c is the cut-off frequency of the low-pass filter, set to 30 Hz. This post-processing step improves the continuity of the estimated velocity, making it more suitable for downstream control and SLAM optimization tasks.

3.4 IMU pre-integration factor

To introduce the IMU pre-integration formulation, we first clarify the coordinate systems and related symbols. Let *{W}* denote the world (inertial) frame and *{B}* the body-fixed frame attached to the robot. The IMU is rigidly mounted on the robot, and its axes are assumed to coincide with the body frame; all other sensor measurements can be transformed into this frame for consistency. Under this assumption, the IMU frame is treated as identical to the body frame.

As shown schematically in Fig. 4, the robot state at time t is represented as shown in Eq. (22).

$$x_t = [R_t^T \quad p_t^T \quad v_t^T \quad b_t^T]^T, \tag{22}$$

where $R_t \in SO(3)$ is the rotation matrix from the body frame *{B}* to the world frame *{W}*; $p_t \in \mathbb{R}^3$ is the position of the IMU or body origin expressed in *{W}* ; $v_t \in \mathbb{R}^3$ is the linear velocity of the body origin in *{W}* ; $b_t = [b_t^\omega, b_t^a]^T$ is the IMU bias vector, including gyroscope bias b_t^ω and accelerometer bias b_t^a.

The raw measurements provided by the IMU consist of angular velocity and linear acceleration (Eq. (23)). They can be modeled as:

$$\begin{cases} \widehat{\omega}_t = \omega_t + b_t^\omega + n_t^\omega, \\ \hat{a}_t = R_t^{BW}(a_t - g) + b_t^a + n_t^a, \end{cases} \tag{23}$$

where $\widehat{\omega}_t$ and $\hat{a}_t$ are the gyroscope and accelerometer outputs expressed in the body frame *{B}*; ω_t and a_t are the true angular velocity and linear acceleration of the body; b_t^ω and b_t^a denote the gyroscope and accelerometer biases, respectively; n_t^ω and n_t^a are zero-mean white noises characterizing sensor measurement errors; g is the gravity vector in the world frame; R_t^{BW} is the rotation matrix from the world frame *{W}* to the body frame *{B}*.

The IMU pre-integration factor uses these measurements to propagate relative motion between keyframes while avoiding repeated integration from the origin at every optimization step. By integrating the bias-corrected angular velocity and acceleration over short time intervals, a compact representation of the relative rotation, velocity, and position is obtained. This formulation greatly improves computational efficiency and allows inertial information to be incorporated as a constraint in the overall factor-graph optimization for SLAM.

Based on the continuous output of linear acceleration and angular velocity from the IMU, the robot's pose at successive time instants can, in principle, be obtained by direct integration of these signals. Using Euler integration, the discrete-time state propagation of the IMU can be written as:

$$\begin{cases} v_{t+\Delta t} = v_t + g\Delta t + R_t(\hat{a}_t - b_t^a - n_t^a)\Delta t, \\ p_{t+\Delta t} = p_t + v_t\Delta t + \frac{1}{2}g\Delta t^2 + \frac{1}{2}R_t(\hat{a}_t - b_t^a - n_t^a)\Delta t^2, \\ R_{t+\Delta t} = R_t\, exp\ ((\widehat{\omega}_t - b_t^\omega - n_t^\omega)\Delta t), \end{cases} \tag{24}$$

Although Eq. (24) is conceptually straightforward, its direct application within a nonlinear optimization framework is computationally demanding. Each time the initial estimate of R_t, v_t, or p_t is updated during optimization, the entire integration from the starting time i to the target time j must be recomputed, because the propagation terms depend explicitly on the absolute orientation R_t. This repeated re-integration severely limits real-time performance, particularly when long IMU sequences are involved.

To address this issue, the IMU pre-integration technique proposed by Forster *et al.*[21] is adopted. The key idea is to separate the inertial measurements from the global state by expressing them in the body frame of the initial keyframe i, and to integrate only the relative motion. Applying this approach to Eq. (24) yields the pre-integrated quantities as shown in Eq. (25).

$$\begin{aligned} \Delta v_{ij} &= R_i^T(v_j - v_i - g\Delta t_{ij}), \\ \Delta p_{ij} &= R_i^T(p_j - p_i - v_i\Delta t_{ij} - \frac{1}{2}g\Delta t_{ij}^2), \\ \Delta R_{ij} &= R_i^T R_j, \end{aligned} \tag{25}$$

where Δv_{ij} , Δp_{ij} , and ΔR_{ij} denote the relative velocity, position, and orientation of the IMU between states *i* and *j*; Δt_{ij} is the elapsed time between the two states.

These pre-integrated terms encapsulate the relative motion of the IMU over the interval [*i, j*]. Crucially, they are independent of the initial state R_i, v_i, and p_i remain fixed during optimization, unless the IMU bias estimates are modified. As a result, when the optimization algorithm, such as a factor-graph solver, refines the global trajectory, it only needs to recompute the pre-integration if the bias changes, rather than repropagating all raw measurements. This significantly improves computational efficiency and makes the fusion of high-rate inertial data with LiDAR or leg odometry feasible in real time.

3.5 LiDAR odometry factor

When a new frame of LiDAR data is received, the first step is to evaluate the local geometric structure of the point cloud in order to extract meaningful features. Following the feature extraction method proposed by Shan *et al.*,[32] curvature is computed for each point in its neighborhood: points with high curvature are labeled as edge features, while those with low curvature are classified as planar features. Let the extracted edge and planar feature sets from the LiDAR scan at time *i* be denoted by F_i^e and F_i^p, respectively. The feature frame can then be expressed as shown in Eq. (26).

$$F_i = \{F_i^e, F_i^p\}, \tag{26}$$

where F_i represents the LiDAR frame at time *i* in the body coordinate system *{B}*.

Directly incorporating every LiDAR frame into the factor graph is computationally prohibitive. To ensure efficiency, a keyframe strategy is applied: a new scan is selected as a keyframe whenever the robot's motion relative to the last keyframe exceeds predefined thresholds. In this study, the thresholds are set to 1 m in translation and 10° in rotation. Once a new keyframe F_{i+1} is selected, the associated robot state x_{i+1} is added to the factor graph, while intermediate frames between F_i and F_{i+1} are discarded. This mechanism yields a sparser factor graph, balancing computational efficiency, map density, and memory usage.

After adding the new state node x_{i+1}, a LiDAR odometry factor is constructed to constrain it relative to x_i. The generation of this factor consists of three steps:

Step 1: Submap construction. A local submap is created using a sliding window of the most recent n keyframes. The feature point sets $\{F_{i-n}, \dots, F_i\}$ are transformed into the world frame using their corresponding poses $\{T_{i-n}, \dots, T_i\}$ and then merged to form a voxelized submap M_i As F_i consists of both edge and planar features, M_i is similarly divided into two voxel maps, M_i^e and M_i^p, defined as shown in Eq. (27).

$$\begin{gathered} M_i = \{M_i^e, M_i^p\}, \\ M_i^e = F_i^e \cup F_{i-1}^e \cup \dots \cup F_{i-n}^e, \end{gathered} \tag{27}$$

where all feature points are expressed in the world frame *{W}*. In this work, *n* = 25 is adopted. Voxel downsampling is applied to M_i^e and M_i^p with resolutions of 0.2 m and 0.4 m, respectively, to reduce redundancy and remove duplicate points.

Step 2: Scan matching. An initial relative transformation $\tilde{T}_{i+1}$ is estimated using IMU pre-integration at time *i* + 1. The point cloud F_{i+1} is transformed into the world frame and matched against the submap M_i. For each edge point in F_{i+1}^e and each planar point in F_{i+1}^p, the corresponding geometric structures, such as lines or planes, are searched within M_i^e and M_i^p, following the scan-matching approach described by Zhang and Singh.[31]

Step 3: Relative transformation estimation. If the pose estimate is correct, edge points should lie on their corresponding edges and planar points should lie on their corresponding surfaces. In practice, misalignments occur, and the distances from points to their corresponding structures are used as residuals as shown in Eq. (28) and Eq. (29).

$$\textstyle\sum_k d_{e_k} + \sum_k d_{p_k}, \tag{28}$$

where

$$\begin{cases} d_{e_k} = \dfrac{|(p_{i+1,k}^e - p_{i,u}^e) \times (p_{i+1,k}^e - p_{i,v}^e)|}{|p_{i,u}^e - p_{i,v}^e|}, \\ d_{p_k} = \dfrac{|(p_{i+1,k}^p - p_{i,u}^p) \cdot ((p_{i,u}^p - p_{i,v}^p) \times (p_{i,u}^p - p_{i,w}^p))|}{|(p_{i,u}^p - p_{i,v}^p) \times (p_{i,u}^p - p_{i,w}^p)|}. \end{cases} \tag{29}$$

where, d_{e_k} and d_{p_k} denote the distances from edge and planar points to their respective structures. $p_{i+1,k}^e$ is an edge point from F_{i+1}^e, while $p_{i,u}^e$ and $p_{i,v}^e$ are two neighboring points on the corresponding line in M_i^e. Similarly, $p_{i+1,k}^p$ is a planar point from F_{i+1}^p, and $\{p_{i,u}^p, p_{i,v}^p, p_{i,w}^p\}$ are three neighboring points defining the corresponding plane in M_i^p.

Nonlinear optimization is then carried out using the Gauss-Newton method to minimize the residuals as shown in Eq. (30).

$$\min_{T_{i+1}} \left\{ \sum_{p_{i+1,k}^e \in F_{i+1}^e} d_{e_k} + \sum_{p_{i+1,k}^p \in F_{i+1}^p} d_{p_k} \right\} \tag{30}$$

The solution provides the relative transformation between the two states as shown in Eq. (31).

$$\Delta T_{i,i+1} = T_i^T T_{i+1}. \tag{31}$$

Finally, this relative transformation is introduced as a LiDAR odometry factor into the factor graph, establishing a geometric constraint between nodes x_i and x_{i+1}. This factor enriches the graph with high-accuracy LiDAR-derived pose information, complementing the IMU and leg odometry factors in the overall multi-sensor fusion framework.

3.6 Loop closure detection factor

Loop closure detection is the process by which a robot recognizes that it has revisited a previously observed location, thereby enabling the correction of accumulated drift and the closure of the global map. Within the factor graph framework, loop closure constraints can be naturally introduced as additional factors, providing long-range consistency to the pose graph and significantly improving global accuracy.

In this study, loop closure detection is implemented primarily using a Euclidean distance-based search combined with geometric verification through scan matching. When a new robot state x_{i+1} is added to the factor graph via a LiDAR keyframe F_{i+1}, the system first searches for candidate loop closure nodes by computing the Euclidean distance between x_{i+1} and all prior robot states. The closest candidate state is denoted x_2. To improve robustness, not only the keyframe F_2 but also its local neighborhood $\{F_{2-m}, \dots, F_2, \dots, F_{2+m}\}$ is retrieved. All these frames are transformed into the world frame $\{W\}$ using their corresponding poses.

Next, the scan matching procedure described in the previous section is applied between F_{i+1} and the submap constructed from the neighboring frames $\{F_{2-m}, \dots, F_2, \dots, F_{2+m}\}$. If a reliable geometric alignment is found, a relative transformation $\Delta T_{2,i+1}$ is obtained. This transformation is then introduced into the factor graph as a loop closure factor, linking the new node x_{i+1} with the historical node x_2.

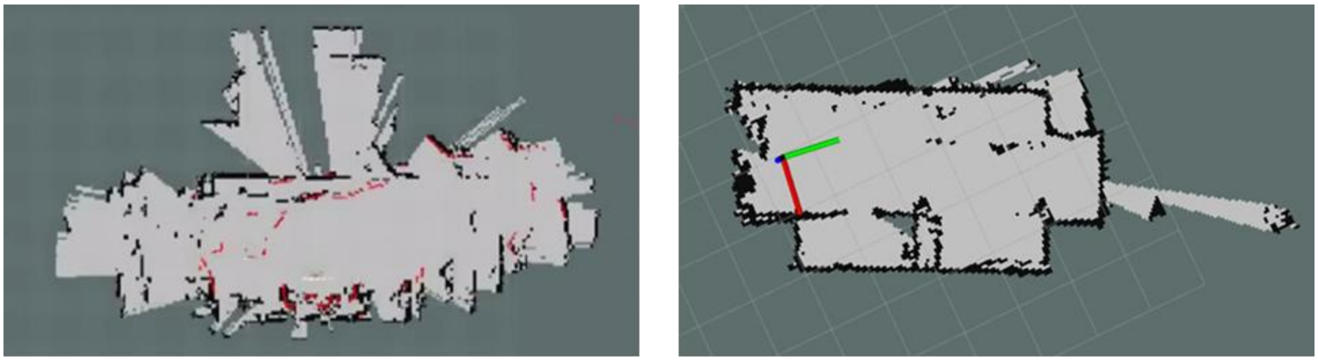
In the practical implementation of this work, the neighborhood parameter is set to $m = 8$, and the maximum search radius for loop closure candidates is 10 m. This configuration strikes a balance between detection sensitivity and computational efficiency, ensuring that loop closure constraints are introduced only when strong geometric evidence supports the match. By incorporating these factors, the system effectively reduces long-term drift and improves the global consistency of the estimated trajectory and reconstructed map.

4. Experiments

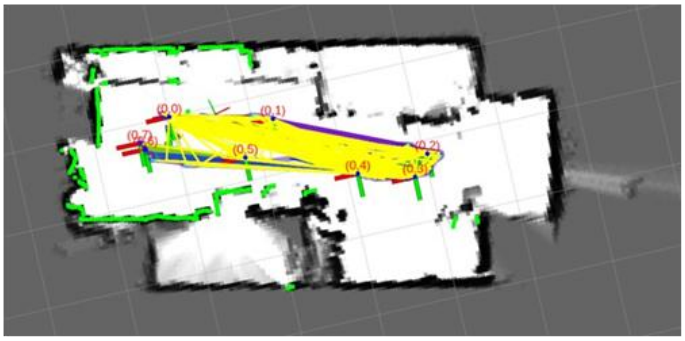
4.1 Single-sensor SLAM

To evaluate the performance of single-modality SLAM methods, mapping experiments were conducted using both laser- based and vision-based algorithms independently. Since many classical algorithms are designed for relatively structured and low-feature environments, indoor scenes were selected to ensure comparability and repeatability. After installing the necessary software dependencies, each SLAM algorithm was initialized individually, and the quadruped robot was manually controlled to navigate within the indoor space while generating maps.

Laser-based SLAM: The mapping results of three representative laser-based algorithms: Gmapping, Hector SLAM, and Cartographer, are shown in Fig. 5. In these grid-style maps, thick black lines denote the boundaries of walls, while white regions correspond to free space. As expected, because only a LiDAR sensor was used, image information is absent in these maps.

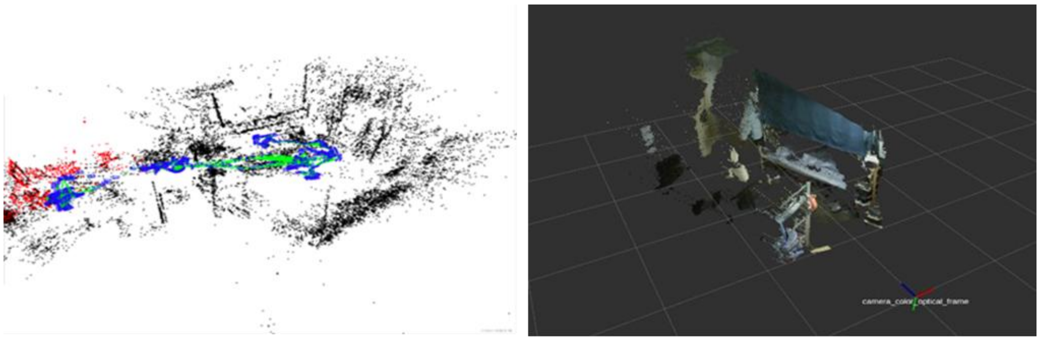


(a) Gmapping mapping (b) Hector SLAM mapping



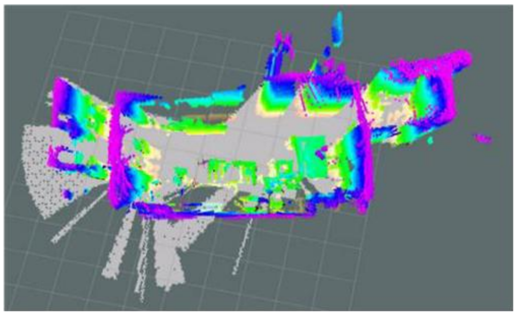
(c) Cartographer mapping

Fig. 5: Mapping results of three laser-based SLAM algorithms.



(a) ORB-SLAM mapping.

(b) RGB-D-SLAM mapping.



(c) RTAB-MAP mapping.

Fig. 6: Mapping results of three visual-based SLAM algorithms.

Vision-based SLAM: Fig. 6 shows the maps constructed by ORB-SLAM, RGB-D-SLAM, and RTAB-MAP. These maps are generally more limited in field of view because they rely on a single camera. ORB-SLAM, which is feature-point based, generates sparse maps that are sufficient for localization but not for dense reconstruction. RGB-D-SLAM provides denser maps, but its limited depth camera field of view causes data loss. RTAB-MAP generates relatively complete maps, yet suffers from noticeable drift and lower real-time performance.

Table 2: Comparison of mapping accuracy for three laser slam algorithms.

Algorithm	Mean Absolute Error (/m)	Standard Deviation (/m)	Maximum Error (/m)	Mean Absolute Relative Error (%)
Gmapping	0.56	0.31	1.22	18.93
Hector SLAM	0.43	0.25	1.03	14.06
Cartographer	0.32	0.19	0.84	10.21

Quantitative Evaluation: For grid maps generated by laser-based SLAM (Fig. 5), the vertical dimension of the indoor environment was chosen as the evaluation metric, since drift and divergence along the horizontal axis made width estimation unreliable. A laser rangefinder was used to measure the ground-truth indoor width, and the same locations were measured within the generated maps using *rviz*. The accuracy results are shown in Table 2. Among the three algorithms, Gmapping underestimated the true distance, while Hector SLAM and Cartographer slightly overestimated. Even the best-performing Cartographer still produced a relative error exceeding 10%, highlighting the sensitivity of LiDAR-only approaches to vibrations induced by the quadruped robot's gait.

For visual SLAM maps, where complete contours were unavailable, the evaluation relied on distances between selected pairs of feature points that were identifiable both in the map and in the real environment. The results in Table 3 indicate that the accuracy of visual SLAM algorithms was generally lower than that of the LiDAR-based algorithms, with relative errors reaching up to 25% for RGB-D-SLAM. This discrepancy compared with reported performances in the literature suggests that robot-induced vibrations significantly degrade visual perception quality.

Table 3: Comparison of mapping accuracy for three visual slam algorithms.

Algorithm	Mean Absolute Error (/m)	Standard Deviation (/m)	Maximum Error (/m)	Mean Absolute Relative Error (%)
ORB-SLAM	0.21	0.19	0.58	11.91
RGB-D-SLAM	0.39	0.27	0.75	25.12
RTAB-MAP	0.30	0.22	0.66	17.54

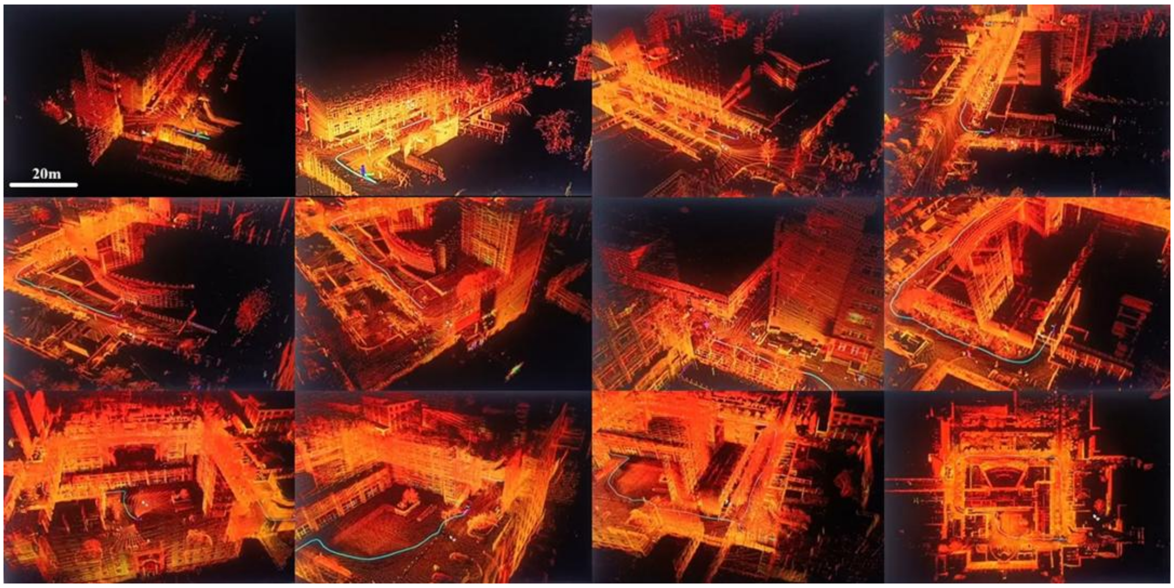


Fig. 7: Mapping process with the proposed multi-sensor fusion algorithm.

Overall, both laser-based and visual-based SLAM algorithms demonstrate distinct advantages and limitations, largely stem- ming from the inherent sensing modalities. However, when deployed on quadruped robots, the mapping accuracy of both approaches degrades significantly due to platform-induced vibrations and dynamic gait motion. These results underscore that single-sensor SLAM is insufficient for achieving high-quality mapping on legged platforms. Consequently, multi-sensor fusion is essential to mitigate individual sensor weaknesses and improve robustness in real-world environments.

4.2 Multi-sensor fusion SLAM

In this section, practical mapping experiments were carried out using the proposed fusion algorithm. Unlike the narrow indoor scenes used in single-sensor experiments, the fusion algorithm relies on extracting a sufficient number of edge and planar features, which requires richer structural information. Therefore, the area surrounding the Mechanical Building at Beijing Jiaotong University was selected as the experimental environment. The quadruped robot was commanded to navigate around the building while recording synchronized LiDAR and IMU data. Graph optimization during mapping was performed using the GTSAM library. The mapping process is illustrated in Fig. 7.

As shown in Fig. 7, the robot started at the southeast corner of the Mechanical Building, circumnavigated the building, and successfully returned to its initial position. The resulting map formed a complete closed loop without tilt or misalignment of ground and wall structures, indicating robust global consistency and high mapping quality. A visual comparison between the reconstructed maps and the real environment is shown in Fig. S1.

From Fig. S1, it can be seen that the fusion algorithm achieved highly faithful 3D reconstructions of the buildings. To further evaluate its adaptability, additional experiments were performed in different environments, including semi-indoor, cluttered outdoor, and tunnel-like scenes (Fig. S2). The results show that the algorithm consistently produced stable maps even under varying terrain and structural constraints.

In semi-indoor environments (Fig. S2(a)), columns and pedestrians were reconstructed with high fidelity, demonstrating robustness in moderately constrained spaces. In cluttered outdoor scenes with trees and overhead obstacles (Fig. 9(c)), as well as tunnel-like environments with limited clearance (Fig. S2(e)), the fusion algorithm also produced stable maps with well-preserved structural details.

Qualitative evaluation: To further validate mapping quality, the reconstructed point cloud was transformed into a top-down view. After removing ground and non-structural elements, the resulting building outlines were aligned with satellite imagery and 2D architectural maps. As shown in Fig. S3, the point cloud maps generated by the fusion algorithm match closely with both reference maps, confirming the system's global consistency.

Algorithmic comparison: For a fair comparison, the improved fusion algorithm and the baseline LeGO-LOAM were tested on the same dataset. As shown in Fig. S4, the fusion algorithm maintained accurate global mapping without drift, while LeGO-LOAM suffered from rapid drift accumulation once the robot began moving. This difference highlights the necessity of tightly integrating IMU and legodometry constraints in addition to LiDAR.

Platform comparison: To investigate the effect of motion-induced vibrations, mapping experiments were also conducted using a wheeled robot along the same trajectory. As shown in Fig. S5, both platforms successfully generated maps, but the wheeled robot achieved slightly smoother reconstructions with fewer structural artifacts. This can be attributed to reduced vibration compared to the quadruped gait.

Table 4: Mapping accuracy results with wheeled robot.

Number	True Value /m	Map Measurement /m	Absolute Error /m	Relative Error %
1	22.00	22.12	0.12	0.55
2	82.00	82.33	0.33	0.40
3	25.00	25.10	0.10	0.40
4	20.00	20.06	0.06	0.30
5	60.00	60.25	0.25	0.42

Table 5: Mapping accuracy results with quadruped robot.

Number	True Value /m	Map Measurement /m	Absolute Error /m	Relative Error %
1	22.00	23.73	1.73	7.86
2	82.00	83.64	1.64	2.00
3	25.00	26.41	1.41	5.64
4	20.00	20.78	0.78	3.90
5	60.00	62.17	2.17	3.62

Quantitative evaluation: To quantify mapping accuracy, reference measurements of the Mechanical Building were obtained from architectural drawings and satellite data (Fig. S6). Using PCL tools, distances in the reconstructed point clouds were compared against ground-truth values. Results for the wheeled and quadruped platforms are summarized in Tables 4 and 5.

While the wheeled robot achieved sub-meter accuracy with errors below 1%, the quadruped robot exhibited slightly larger errors (up to 7.86%), primarily due to locomotion-induced vibrations.

Compared with single-sensor SLAM methods, the proposed fusion algorithm not only enables fine-grained 3D point cloud mapping by tightly combining LiDAR, IMU, and leg odometry data, but also mitigates vibration-induced errors inherent to quadruped locomotion. Overall mapping accuracy reached approximately 4.6% in terms of mean relative absolute error (MRAE) across all measured segments, significantly outperforming single-sensor baselines.

5. Discussion

The experimental results presented above highlight both the advantages and limitations of different SLAM strategies when applied to quadruped robots. Several key observations can be drawn:

Impact of platform vibrations: A consistent finding across all experiments is the significant influence of quadruped locomotion on SLAM performance. Both LiDAR- and vision-based single-sensor methods suffered from noticeable drift and mapping artifacts, with relative errors ranging from 10% to over 25%. This indicates that algorithms originally developed for wheeled robots cannot be directly transferred to legged platforms without accounting for motion-induced disturbances.

Single-sensor limitations: While Cartographer and ORB-SLAM emerged as the best-performing algorithms in their respective categories, their accuracy was still insufficient for high-fidelity mapping in quadruped scenarios. LiDAR-based approaches produced clearer global structures but were vulnerable to vibration, whereas vision-based methods provided richer local textures but suffered from a limited field of view and greater sensitivity to motion blur. These complementary weaknesses underline the necessity of multi-sensor fusion.

Effectiveness of the proposed fusion algorithm: By tightly coupling LiDAR, IMU, and leg odometry through factor graph optimization, the proposed system successfully mitigated the drift and distortion observed in unimodal approaches. The algorithm consistently produced globally consistent 3D point cloud maps in both structured (semi-indoor) and unstructured (outdoor cluttered or tunnel-like) environments. Quantitatively, the fusion method reduced the mean relative absolute error to approximately 4.6% on the quadruped robot, a substantial improvement over the 10-25% error range of single-sensor SLAM.

Comparison with wheeled robots: When tested on a wheeled robot, the same fusion algorithm achieved sub-meter accuracy with relative errors below 1%. This contrast emphasizes that while the algorithm is robust, the physical platform still plays a decisive role in determining mapping quality. Quadruped vibrations remain a fundamental challenge that cannot be entirely eliminated by software-level improvements.

Remaining challenges: Despite these improvements, several limitations remain. First, the system's reliance on feature- rich environments means that performance in extremely narrow or textureless indoor corridors is still suboptimal. Second, computational cost increases with multi-sensor fusion, which may limit real-time deployment on resource-constrained platforms. Finally, while loop closure detection improves global consistency, it may still fail under highly dynamic conditions where environmental features change rapidly.

In summary, the discussion highlights that multi-sensor fusion provides a viable path toward robust SLAM on quadruped robots, but challenges related to locomotion vibrations, environmental constraints, and computational efficiency remain. These insights provide the basis for the concluding remarks and directions for future research.

6. Conclusion

In this paper, we investigated the challenges of SLAM for legged robots, particularly the vibration-induced mapping errors caused by quadruped locomotion, and proposed a multi-sensor fusion framework to address them. The main contributions and findings can be summarized as follows:

Experimental platform: A universal experimental platform was developed, consisting of a Unitree Go1 quadruped robot equipped with a 16-line LiDAR, an RGB-D camera, and a ten-axis IMU. This platform enabled fair and repeatable comparative studies across multiple sensor modalities and SLAM algorithms.

Baseline evaluation with single sensors: Extensive experiments with six representative single-sensor SLAM algorithms demonstrated the limitations of unimodal perception under quadruped motion. - For LiDAR-based SLAM, Cartographer outperformed Hector SLAM and Gmapping, but still exhibited an average relative error of more than 10%. - For vision- based SLAM, ORB-SLAM achieved the best performance among the tested methods, but relative errors exceeded 12%, and robustness was compromised by motion-induced vibrations. These results confirm that vibrations significantly degrade both LiDAR- and vision-only SLAM on quadruped robots.

Proposed fusion algorithm: To overcome these limitations, we proposed an improved LiDAR-inertial-leg odometry (LILO) algorithm. Based on the LeGO-LOAM framework, IMU/leg pre-integration factors were incorporated into the optimization graph to correct distorted point clouds and provide drift-free velocity priors during stance phases. The fusion framework was implemented using GTSAM for factor graph optimization.

Experimental validation: The proposed algorithm was evaluated in diverse environments, including semi-indoor spaces, cluttered outdoor areas, and tunnel-like passages. Qualitative results showed high-fidelity 3D reconstruction with closed- loop maps and globally consistent alignment with satellite imagery and 2D building maps. Quantitatively, the fusion algorithm achieved a mean relative absolute error (MRAE) of approximately 4.6% across all measurement segments on the quadruped robot, which is a substantial improvement over single-sensor SLAM (10-25% errors). On a wheeled robot, the error was further reduced to below 1%, highlighting the effect of locomotion vibrations on mapping accuracy.

Overall, the experimental results demonstrate that the proposed multi-sensor fusion SLAM framework effectively mitigates vibration-induced degradation and significantly enhances the accuracy and robustness of quadruped robot mapping. Compared with state-of-the-art single-sensor baselines, the proposed system achieves higher mapping fidelity, more stable loop closure, and improved adaptability to diverse environments.

Future work will focus on extending this framework to collaborative multi-robot SLAM in complex outdoor environments, exploring heterogeneous robot teams, and incorporating learning-based modules to further enhance robustness in unstructured and dynamic scenarios.

Acknowledgments

This work was supported by Tangshan City Applied Basic Research Science and Technology Plan Project (25130201B), Beijing Natural Science Foundation (L243013), Central Government Guides Local Science and Technology Development Fund Projects (246Z1813G), National Key Research and Development Program of China (2022YFB4701600), and National Natural Science Foundation of China (62103036).

Conflict of Interest

There is no conflict of interest.

Supporting Information

Applicable.

CRediT Statement

Guangrong Chen: Conceptualization, Funding acquisition, Methodology, Project administration, Resources, Supervision, Writing – Original draft, Writing – Review & editing. **Qizhe Yang:** Data curation, Formal analysis, Investigation, Software, Validation, Visualization. **Mengqi Yang:** Data curation, Formal analysis, Investigation, Software, Validation, Visualization. **Guangxin Zhang:** Data curation, Formal analysis, Investigation, Software, Validation, Visualization. **Mengqiu Mo:** Data curation, Formal analysis, Investigation, Software, Validation, Visualization. **Yuxiang Lin:** Data curation, Formal analysis, Investigation, Software, Validation, Visualization.

References

[1] R. Giubilato, W. Stürzl, A. Wedler, R. Triebel, Challenges of SLAM in extremely unstructured environments: the DLR planetary stereo, solid-state LiDAR, inertial dataset, *IEEE Robotics and Automation Letters*, 2022, **7**, 8721-8728, doi: 10.1109/LRA.2022.3188118.

[2] X. Xu, L. Zhang, J. Yang, C. Cao, W. Wang, Y. Ran, Z. Tan, M. Luo, A review of multi-sensor fusion SLAM systems based on 3D LIDAR, *Remote Sensing*, 2022, **14**, 2835, doi: 10.3390/rs14122835.

[3] L. Gao, Y. Liang, J. Yang, S. Wu, C. Wang, J. Chen, L. Kneip, VECtor: a versatile event-centric benchmark for multi-sensor SLAM, *IEEE Robotics and Automation Letters*, 2022, **7**, 8217-8224, doi: 10.1109/LRA.2022.3186770.

[4] C. Campos, R. Elvira, J. J. G. Rodríguez, J. M. M. Montiel, J. D. Tardós, ORB-SLAM3: an accurate open-source library for visual, visual–inertial, and multimap SLAM, *IEEE Transactions on Robotics*, 2021, **37**, 1874-1890, doi: 10.1109/TRO.2021.3075644.

[5] X. Gao, Research on indoor robot mapping and navigation based on multi-sensor fusion, Ph.D. dissertation, Ningxia University, 2021.

[6] J. Zhang, S. Singh, LOAM: lidar odometry and mapping in real-time, *Robotics: Science and Systems Foundation*, 2014, doi: 10.15607/rss.2014.x.007.

[7] T. Shan, B. Englot, D. Meyers, W. Wang, C. Ratti, D. Rus, LIO-SAM: tightly-coupled lidar inertial odometry *via* smoothing and mapping, *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS),* Vegas, NV, USA, October 24 2020-January 24, 2021, Las 5135-5142, doi: 10.1109/iros45743.2020.9341176.

[8] W. Xu, F. Zhang, FAST-LIO: a fast, robust LiDAR-inertial odometry package by tightly-coupled iterated Kalman filter, *IEEE Robotics and Automation Letters*, 2021, **6**, 3317-3324, doi: 10.1109/LRA.2021.3064227.

[9] W. Xu, Y. Cai, D. He, J. Lin, F. Zhang, FAST-LIO2: fast direct LiDAR-inertial odometry, *IEEE Transactions on Robotics*, 2022, **38**, 2053-2073, doi: 10.1109/TRO.2022.3141876.

[10] H. Ye, Y. Chen, M. Liu, Tightly coupled 3D lidar inertial odometry and mapping, *International Conference on Robotics and Automation (ICRA)*, Montreal, QC, Canada, May 20-24, 2019, 3144-3150, doi: 10.1109/icra.2019.8793511.

[11] D. Wisth, M. Camurri, M. Fallon, VILENS: visual, inertial, lidar, and leg odometry for all-terrain legged robots, *IEEE Transactions on Robotics*, 2022, **39**, 309-326, doi: 10.1109/TRO.2022.3193788.

[12] T. Miki, J. Lee, J. Hwangbo, L. Wellhausen, V. Koltun, M. Hutter, Learning robust perceptive locomotion for quadrupedal robots in the wild, *Science Robotics*, 2022, **7**, eabk2822, doi: 10.1126/scirobotics.abk2822.

[13] G. Chen, L. Hong, Research on environment perception system of quadruped robots based on LiDAR and vision, *Drones*, 2023, **7**, 329, doi: 10.3390/drones7050329.

[14] J. Zhou, Research and implementation of autonomous exploration and mapping system for mobile robot based on multi-sensor, Ph.D. dissertation, Nanjing University of Posts and Telecommunications, 2020.

[15] J. Li, H. Qin, J. Wang, J. Li, OpenStreetMap-based autonomous navigation for the four wheel-legged robot *via* 3D-lidar and CCD camera, *IEEE Transactions on Industrial Electronics*, 2022, **69**, 2708-2717, doi: 10.1109/TIE.2021.3070508.

[16] Z. Zhu, J. Wang, M. Xu, Z. Chen, Rein-SLAM: narrow the gaps between the matching task and SLAM system, *IEEE Transactions on Industrial Electronics*, 2023, **70**, 10353-10362, doi: 10.1109/TIE.2022.3219119.

[17] T. Qin, P. Li, S. Shen, VINS-mono: a robust and versatile monocular visual-inertial state estimator, *IEEE Transactions on Robotics*, 2018, **34**, 1004-1020, doi: 10.1109/TRO.2018.2853729.

[18] A. Li, J. Wang, M. Xu, Z. Chen, DP-SLAM: a visual SLAM with moving probability towards dynamic environments, *Information Sciences*, 2021, **556**, 128-142, doi: 10.1016/j.ins.2020.12.019.

[19] Y.-J. Ha, K.-H. Kim, B. W. Nam, S. Y. Hong, H. Kim, Experimental study for characteristics of slamming loads on bow of a ship-type FPSO under breaking and irregular wave conditions, *Ocean Engineering*, 2021, **224**, 108738, doi: 10.1016/j.oceaneng.2021.108738.

[20] G. Li, Y. Zeng, H. Huang, S. Song, B. Liu, X. Liao, A multi-feature fusion slam system attaching semantic invariant to points and lines, *Sensors*, 2021, **21**, 1196, doi: 10.3390/s21041196.

[21] C. Forster, L. Carlone, F. Dellaert, D. Scaramuzza, On-manifold preintegration for real-time visual: inertial odometry, *IEEE Transactions on Robotics*, 2017, **33**, 1-21, doi: 10.1109/TRO.2016.2597321.

[22] E. Besada-Portas, J. A. Lopez-Orozco, J. Besada, J. M. de la Cruz, Multisensor fusion for linear control systems with asynchronous, Out-Of-Sequence and erroneous data, *Automatica*, 2011, **47**, 1399-1408, doi: 10.1016/j.automatica.2011.02.030.

[23] G. G. Rigatos, Extended Kalman and Particle Filtering for sensor fusion in motion control of mobile robots, *Mathematics and Computers in Simulation*, 2010, **81**, 590-607, doi: 10.1016/j.matcom.2010.05.003.

[24] S. Li, X. Li, S. Chen, Y. Zhou, S. Wang, Two-step LiDAR/camera/IMU spatial and temporal calibration based on continuous-time trajectory estimation, *IEEE Transactions on Industrial Electronics*, 2024, **71**, 3182-3191, doi: 10.1109/TIE.2023.3270506.

[25] H. Li, B. Tian, H. Shen, J. Lu, An intensity-augmented LiDAR-inertial SLAM for solid-state LiDARs in degenerated environments, *IEEE Transactions on Instrumentation and Measurement*, 2022, **71**, 8503610, doi: 10.1109/TIM.2022.3190060.

[26] L. Xu, H. Yin, T. Shi, D. Jiang, B. Huang, EPLF-VINS: real-time monocular visual-inertial SLAM with efficient point-line flow features, *IEEE Robotics and Automation Letters*, 2023, **8**,

752-759, doi: 10.1109/LRA.2022.3231983.
[27] J. Chen, H. Wang, M. Hu, P. N. Suganthan, Versatile LiDAR-inertial odometry with SE(2) constraints for ground vehicles, *IEEE Robotics and Automation Letters*, 2023, **8**, 3486-3493, doi: 10.1109/LRA.2023.3268584.
[28] Y. Zhang, LILO: a novel lidar–IMU SLAM system with loop optimization, *IEEE Transactions on Aerospace and Electronic Systems*, 2021, **58**, 2649-2659, doi: 10.1109/TAES.2021.3135234.
[29] L. Cao, X. Xiao, A visual-inertial SLAM method based on rolling shutter RGB-D cameras, *Robot*, 2021, **43**, 193–202, doi: 10.13973/j.cnki.robot.200245.
[30] X. Gong, H. Min, Indoor positioning by visual-inertial odometry based on ROS, *Modular Machine Tool & Automatic Manufacturing Technique*, 2019, **5**, 106–110, doi: 10.13462/j.cnki.mmtamt.2019.05.026.
[31] J. Zhang, S. Singh, Low-drift and real-time lidar odometry and mapping, *Autonomous Robots*, 2017, **41**, 401-416, doi: 10.1007/s10514-016-9548-2.
[32] T. Shan, B. Englot, LeGO-LOAM: lightweight and ground-optimized lidar odometry and mapping on variable terrain, *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Madrid, Spain, October 1-5, 2018, 4758-4765, doi: 10.1109/IROS.2018.8594299.
[33] J. Lin, F. Zhang, Loam livox: a fast, robust, high-precision LiDAR odometry and mapping package for LiDARs of small FoV, *IEEE International Conference on Robotics and Automation (ICRA)*, Paris, France, May 31-August 31, 2020, 3126-3131., doi: 10.1109/icra40945.2020.9197440.