

Leg-KILO: Robust Kinematic-Inertial-Lidar Odometry for Dynamic Legged Robots

Guangjun Ou, Dong Li, and Hanmin Li

Abstract—This paper presents a robust multi-sensor fusion framework, Leg-KILO (Kinematic-Inertial-Lidar Odometry). When lidar-based SLAM is applied to legged robots, high-dynamic motion (e.g., trot gait) introduces frequent foot impacts, leading to IMU degradation and lidar motion distortion. Direct use of IMU measurements can cause significant drift, especially in the z-axis direction. To address these limitations, we tightly couple leg odometry, lidar odometry, and loop closure module based on graph optimization. For leg odometry, we propose a kinematic-inertial odometry using an on-manifold error-state Kalman filter, which incorporates the constraints from our proposed contact height detection to reduce height fluctuations. For lidar odometry, we present an adaptive scan slicing and splicing method to alleviate the effects of high-dynamic motion. We further propose a robot-centric incremental mapping system that enhances map maintenance efficiency. Extensive experiments are conducted in both indoor and outdoor environments, showing that Leg-KILO has lower drift performance compared to other state-of-the-art lidar-based methods, especially during high-dynamic motion. To benefit the legged robot community, a lidar-inertial dataset containing leg kinematic data and the code are released.

Index Terms—Legged Robots, Localization, SLAM, Sensor Fusion.

I. INTRODUCTION

REAL-TIME state estimation is a crucial component in the navigation and control of legged robots. While robots operating in GPS-denied or poor lighting conditions, Lidar-Inertial Odometry (LIO) is a popular solution. Lidar can directly obtain dense environmental information with depth, while the integration of high-frequency IMU measurements can enhance the system's dynamic response. However, due to the limited sampling frequency and the measurement range, the accelerometer often degrades in the presence of large impacts or high-frequency vibrations [1] [2], which might affect the accuracy of pure LIO system. Fig. 3 shows the results of two subsystems of LIO-SAM [3], IMU odometry and LiDAR odometry, on flat ground. Due to the dynamic motion of the legged robot, the IMU odometry accumulates significant errors in a short time.

Manuscript received: March, 22, 2024; Revised May, 17, 2024; Accepted July, 19, 2024.

This paper was recommended for publication by Editor Abderrahmane Kheddar upon evaluation of the Associate Editor and Reviewers' comments. This work was supported by the Guangdong Basic and Applied Basic Research Foundation No. 2021A1515011867. (*Corresponding author: Dong Li.*)

The authors are with the School of Automation, Guangdong University of Technology, Guangzhou 510006, China ouguangjun98@gmail.com, dong.li@gaut.edu.cn, leehanmincloud@gmail.com

Digital Object Identifier (DOI): see top of this page.

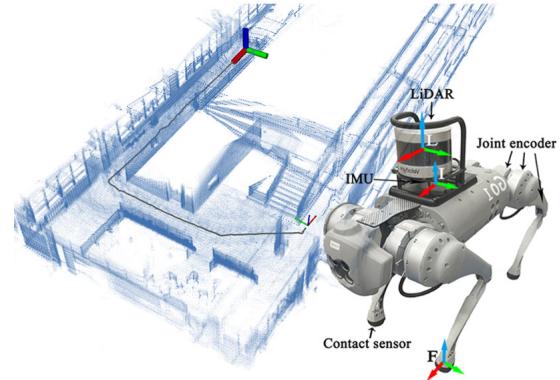


Fig. 1. Left: The map is being built while the legged robot operating across the corridor. Right: Frames and sensors installed on the Unitree Go1. More details are shown in the video at <https://youtu.be/6O74De5BLcQ>

Kim et al. [1] presented an adjustable neck system to absorb the impact in dynamic motion, showing the robustness of the estimation performance. Many previous works [4]–[6] utilized leg kinematic information to provide more constraints for legged robot state estimation. These kinematic-inertial methods still have a large drift under long-term operation. They may fail when a sudden situation occurs (e.g., foot slip) and cannot get enough observations of the yaw angle and the position [4], [7]. The introduction of exteroceptive sensors can effectively address the limitations. Currently many methods [8]–[14] that integrate exteroceptive sensors (e.g. lidar and camera) have been validated in various environments.

However, most of the above methods about legged robots have been tested at a low or moderate speed, and seldom investigated state estimation during high-dynamic motion. In this paper, our aim is to improve the performance of lidar-based SLAM in dynamic legged robots. We propose a tightly couple Kinematic-Inertial-Lidar Odometry framework, Leg-KILO. The system architecture is shown in Fig. 2. Leg-KILO has the following contributions:

- We present a novel leg odometry using an on-manifold error-state Kalman filter (ESKF), tightly coupling leg kinematic and IMU. The height constrains from the contact height detection are fused to reduce the effect of foot impacts. To enhance global localization accuracy, we tightly couple leg odometry, lidar odometry, and loop closure module based on graph optimization.
- To alleviate the effects of high-dynamic motion, we propose an Adaptive Scan Slice and Splice method to

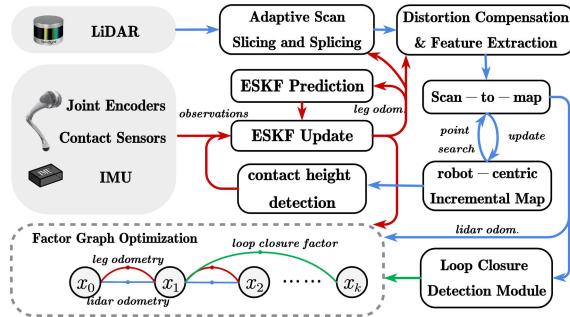


Fig. 2. System overview.

improve the input frequency based on the robot's motion velocity. We further present an incremental robot-centric map approach to enhance map maintenance efficiency, addressing potential heavy computation due to the increased input frequency.

- The experimental result demonstrate that Leg-KILO achieves lower drift compared to other LIO methods. The dataset¹ and the code² are released for the benefit of the community.

II. RELATE WORKS

Multi-sensor fusion is a commonly used solution for robot state estimation. For legged robots, some earlier methods [4]–[6] focused on the use of proprioceptive sensors, such as joint encoders, contact sensors, and IMU, for state estimation. Bloesch et al. [5] proposed an Unscented Kalman Filter (UKF) estimator that added the kinematic velocity to the filter observation. Bledt et al. [6] combines the above observation but utilizes a linear Kalman filter (KF) for estimation. In order to reduce the cumulative error, they also take the observation of contact height with respect to the global frame as a filter measurement, but they set this measurement manually because it cannot be obtained only by proprioceptive sensors. Kim et al. [15] proposed a state estimator capable of utilizing the Gauss-Newton method and handling errors during sudden situations (e.g., foot slip). Hartley et al. [16] uses contact preintegration to build constraints between keyframes, avoiding the excessive introduction of factor graph nodes.

Although the kinematic-inertial methods are low-cost and efficient for state estimation, the absolute position and yaw angle cannot be directly observed [4], [7], leading to significant drift over long periods of movement. Many methods [8]–[13], [16] that further integrate cameras have been validated for their robustness, outperforming kinematic-inertial methods.

Utilizing lidar for state estimation is currently a trending topic in robotics. Shan et al. further improved upon LOAM [17] and proposed a lidar-inertial factor graph framework, LIO-SAM [3]. Xu et al. [18] presented a filter-based lidar-inertial odometry, FAST-LIO2, which is famous for its low-latency. However, above state-of-the-art LIO methods might perform not well when applied to legged robots. This is

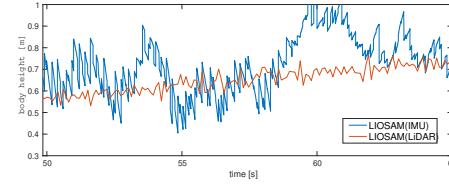


Fig. 3. Comparison of body height's variation for the two subsystems of LIOSAM [3]—IMU odometry and lidar odometry, on sequence *corridor*. Due to the dynamic motion of the legged robot, the IMU odometry accumulates significant errors in a short time.

because the high-frequency impacts of the robot's foot on the ground make accelerometer measurements unstable, resulting in significant drift along z-axis. POINT-LIO [19] is an extension of FASTLIO2 but uses the IMU as a filter measurement rather than an input, and does not rely too much on the IMU measurement. Our experiments also demonstrate its robustness during aggressive motion for legged robots. VILENS [14] comprehensively couples leg kinematics and IMU with lidar and camera to avoid the situation where the degradation of a single sensor leads to incorrect state estimation. VILENS uses graph factor to optimize visual and lidar features to get an accurate state estimation, but runs ICP at a low frequency (2 Hz) due to the heavy computation.

For the lidar-based dataset on legged robots, VILENS [14] collected extensive data on different ANYmal robots, but is not publicly available. The datasets in [20]³ and [21]⁴ are both from the DARPA Subterranean Challenge, but lack raw data about leg kinematics. To verify our Leg-KILO and other lidar-based methods on dynamic legged robots, we conduct extensive data collection including encoders, contact sensors, lidar and IMU, and make our dataset publicly available. More information about our dataset and others on legged robots could be found in Table I.

TABLE I
COMPARISON OF RELATED DATASETS ON LEGGED ROBOTS

Dataset	Robot	Ground Truth	Speed [m/s] mean (max)	Camera	Lidar	IMU	Kin.	Open
VILENS [14]	ANYmal	LT/ICP	0.25 (–)	✓	✓	✓	✓	
Cerberus [20]	ANYmal	ICP	0.36 (0.7)	✓	✓	✓		✓
NeBula [21]	Spot	ICP	0.53 (1.2)	✓	✓	✓		✓
Leg-KILO	Go1	ICP/Slam/ (start end)	0.80 (2.0)		✓	✓	✓	✓

LT: Laser Tracker (start end): Evaluate the drift when returning to the start point.

III. LEG ODOMETRY

A. Filter State Definition

The notation involved in this paper and its explanations are shown in Table II. For leg odometry, we define the filter state $\mathbf{x} \in \text{SO}(3) \times \mathbb{R}^{33}$ as follows:

$$[^W\mathbf{R} \ ^W\mathbf{p} \ ^W\mathbf{v} \ ^W\mathbf{c}_1 \dots \ ^W\mathbf{c}_4 \ ^W\mathbf{g} \ \mathbf{b}_a \ \mathbf{b}_\omega \ \mathbf{a} \ \boldsymbol{\omega}] \quad (1)$$

where $^W\mathbf{R} \in \text{SO}(3)$, $^W\mathbf{p} \in \mathbb{R}^3$, $^W\mathbf{v} \in \mathbb{R}^3$ is the body rotation matrix, position and velocity in the world frame W (i.e., the

¹<https://github.com/ouguangjun/legkilo-dataset>

²<https://github.com/ouguangjun/LegKILO>

³https://github.com/leggedrobotics/cerberus_darpa_subt_datasets

⁴<https://github.com/NeBula-Autonomy/nebula-odometry-dataset>

TABLE II
NOTATIONS

Notation	Explanation
\boxplus / \boxminus	The boxplus and boxminus operators on manifold
$\text{Exp}(\cdot) / \text{Log}(\cdot)$	The map between manifold and tangent vector
$\delta(\cdot)$	The error of the state (\cdot)
$(\cdot)^{\wedge}$	The skew-symmetric matrix
$(\cdot)^{\wedge} / (\cdot)^{\wedge}$	The prior and posterior of state (\cdot)

initial body frame). For easy use of leg kinematics, we add the foot contact points $[{}^W \mathbf{c}_1, {}^W \mathbf{c}_2, {}^W \mathbf{c}_3, {}^W \mathbf{c}_4]$ in the world frame W to the filter state. ${}^W \mathbf{g}$ is the unknown gravity vector in the world frame, which need to be initialization at first. $\mathbf{b}_a, \mathbf{b}_\omega$ are the IMU biases in the body frame. Erroneous measurements from the IMU can result in inaccurate state estimation when IMU is used as input. Given these limitations, we are inspired by [19] to treat the IMU as a model observation, and add the acceleration \mathbf{a} and the angular velocity $\boldsymbol{\omega}$ in the body frame to the filter state.

B. System Kinematics

According to the kinematics of the system, we define the continuous time model of the system at time t as follows:

$$\begin{aligned} {}^W \dot{\mathbf{R}} &= {}^W \mathbf{R} \boldsymbol{\omega}^{\wedge}, & {}^W \dot{\mathbf{p}} &= {}^W \mathbf{v}, \\ {}^W \dot{\mathbf{v}} &= {}^W \mathbf{R} \mathbf{a} + {}^W \mathbf{g}, & {}^W \dot{\mathbf{g}} &= \mathbf{0}, \\ \dot{\mathbf{b}}_a &= \mathbf{n}_a, & \dot{\mathbf{b}}_\omega &= \mathbf{n}_\omega, \\ \dot{\mathbf{a}} &= \mathbf{w}_a, & \dot{\boldsymbol{\omega}} &= \mathbf{w}_\omega \\ {}^W \dot{\mathbf{c}}_i &= \mathbf{w}_{ci} \quad i \in \{1, \dots, 4\} \end{aligned} \quad (2)$$

where $\mathbf{n}_a, \mathbf{n}_\omega$ represents the Gaussian noise of IMU biases. As described in [19], the random walk of the robot's acceleration and angular velocity can be driven by Gaussian noise \mathbf{w}_a and \mathbf{w}_ω , respectively, due to the smoothness of the robot movement. Due to the presence of foot slippage, the random foot position walk can also be described by white noise \mathbf{w}_c , which will be magnified infinitely while the leg swings and reset while the foot makes contact with the ground.

Then, the continuous model (2) is discrete as below:

$$\mathbf{x}_{k+1} = \mathbf{x}_k \boxplus \mathbf{f}(\mathbf{x}_k, \mathbf{w}_k) \quad (3)$$

where k is the index of the filter process and the noise vector \mathbf{w} is defined as $[\mathbf{w}_{c1} \dots \mathbf{w}_{c4} \mathbf{n}_a \mathbf{n}_\omega \mathbf{w}_a \mathbf{w}_\omega]$. $\mathbf{f}(\mathbf{x}, \mathbf{w})$ is the kinematics function:

$$[\mathbf{w} \ {}^W \mathbf{v} \ {}^W \mathbf{R} \mathbf{a} + {}^W \mathbf{g} \ \mathbf{w}_{c1} \dots \mathbf{w}_{c4} \ \mathbf{n}_a \ \mathbf{n}_\omega \ \mathbf{0} \ \mathbf{w}_a \ \mathbf{w}_\omega] \quad (4)$$

C. State Prediction

Suppose the predict process goes from the last time k to the current time $k+1$, the prior state $\hat{\mathbf{x}}_{k+1}$ can be propagated through the formula (3) without considering the noise \mathbf{w} :

$$\hat{\mathbf{x}}_{k+1} = \bar{\mathbf{x}}_k \boxplus \mathbf{f}(\bar{\mathbf{x}}_k, \mathbf{0}) \Delta t. \quad (5)$$

The error state between the ground-truth state and the prior state can be linearized as:

$$\begin{aligned} \delta \mathbf{x}_{k+1} &= \mathbf{x}_{k+1} \boxminus \hat{\mathbf{x}}_{k+1} \\ &\simeq \mathbf{F}_{\delta \mathbf{x}} \delta \mathbf{x}_k + \mathbf{F}_{\mathbf{w}} \mathbf{w}_k \end{aligned} \quad (6)$$

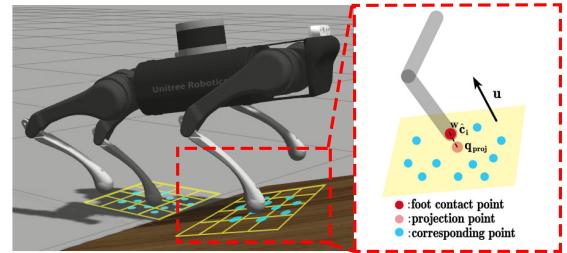


Fig. 4. The contact height detection module.

where $\mathbf{F}_{\delta \mathbf{x}}$ and $\mathbf{F}_{\mathbf{w}}$ are Jacobian matrices of $\mathbf{x} \boxminus \hat{\mathbf{x}}$ with respect to $\delta \mathbf{x}$ and \mathbf{w} , respectively. Since both $\delta \mathbf{x}$ and \mathbf{w} are Gaussian distributions, the prior covariance can be linearized as follows:

$$\hat{\mathbf{P}}_{k+1} = \mathbf{F}_{\delta \mathbf{x}} \bar{\mathbf{P}}_k \mathbf{F}_{\delta \mathbf{x}}^T + \mathbf{F}_{\mathbf{w}} \mathbf{Q}_k \mathbf{F}_{\mathbf{w}}^T \quad (7)$$

where $\bar{\mathbf{P}}$ is the posterior covariance matrix and \mathbf{Q} is the covariance matrix of \mathbf{w} .

D. Measurements

1) *Foot Velocity Measurement*: The foot velocity ${}^W \mathbf{v}_{ci}$ in the world frame can be computed as [5]:

$${}^W \mathbf{v}_{ci} = {}^W \mathbf{v} + {}^W \mathbf{R} (\boldsymbol{\omega}^{\wedge} {}^B \mathbf{c}_i + {}^B \mathbf{v}_{ci}). \quad (8)$$

where ${}^B \mathbf{c}_i = \mathbf{g}_c(\boldsymbol{\alpha}_i)$ and $\mathbf{g}_c(\boldsymbol{\alpha}_i)$ is the leg i forward kinematics function through the joint angle measurement $\boldsymbol{\alpha}_i$. Assuming that there is no relative slip when the i 'th foot makes contact with the ground, the foot velocity measurement can be obtained, which is zero in the ground-truth state:

$$\mathbf{0} = \mathbf{h}_{FV}(\mathbf{x}, \mathbf{n}_{FV}) = {}^W \mathbf{v} + {}^W \mathbf{R} (\boldsymbol{\omega}^{\wedge} {}^B \mathbf{c}_i + {}^B \mathbf{v}_{ci}) + \mathbf{n}_c \quad (9)$$

We define the noise caused by imprecise modeling as \mathbf{n}_c .

2) *Foot Position Measurement*: Given the rotation matrix, body position, and foot position in the world frame, the foot position in the body frame can be calculated as follows:

$${}^B \mathbf{c}_i = \mathbf{h}_{FP}(\mathbf{x}, \mathbf{n}_{FP}) = {}^W \mathbf{R}^T ({}^W \mathbf{c}_i - {}^W \mathbf{p}) + \mathbf{n}_{FP} \quad (10)$$

where \mathbf{n}_{FP} is the Gaussian noise included in the measurement model.

3) *Contact Height Measurement*: The introduction of contact height measurement can help to reduce drift along the Z-axis of the robot [6]. This paper proposes a contact height detection, which involves measuring contact height by utilizing a robot-centric incremental map constructed from historical point clouds. When foot i makes contact with the ground at time t_n , the foot position ${}^B \mathbf{c}_i = \mathbf{g}_c(\boldsymbol{\alpha}_i)$ that can be transformed to the world frame as follows:

$${}^W \hat{\mathbf{c}}_i = \bar{\mathbf{T}}_{f,m} \bar{\mathbf{T}}_{leg,m}^{-1} \hat{\mathbf{T}}_{leg,n} \mathbf{g}_c(\boldsymbol{\alpha}_i) \quad (11)$$

where $\bar{\mathbf{T}}_{f,m}$ represents the transformation optimized by the factor graph (see Section V) at the nearest past time t_m ($t_m < t_n$), while $\bar{\mathbf{T}}_{leg,m}$ and $\hat{\mathbf{T}}_{leg,n}$ represent the transformation of the leg odometry at time t_m and t_n , respectively. The incorporation of the optimized state from factor graph optimization to predict foot position is aimed at mitigating

the effects of long-term cumulative errors in leg odometry. We assume that the local ground around the foot i (within 10 cm distance) is flat at the time of contact, and the foot i lies on a local small plane. This assumption is reasonable in a wider range of paved surfaces, such as cement paths, paved floors, and slopes. We first utilize the foot position ${}^W\hat{\mathbf{c}}_i$ to search its ten corresponding points \mathcal{P}_c (within 10 cm distance) through the robot-centric incremental map (see Section IV B), then the mean $\bar{\mathbf{q}}$ and the covariance \mathbf{A} of the corresponding points \mathcal{P}_c can be defined as follows:

$$\begin{aligned}\mathbf{A} &= \frac{1}{N} \sum_{k=1}^N (\mathbf{q}_k - \bar{\mathbf{q}})(\mathbf{q}_k - \bar{\mathbf{q}})^T \\ \bar{\mathbf{q}} &= \frac{1}{N} \sum_{k=1}^N \mathbf{q}_k, \quad \mathbf{q}_k \in \mathcal{P}_c.\end{aligned}\quad (12)$$

According to the principles of Principal Component Analysis (PCA), the unit normal vector \mathbf{u} of a plane can be obtained by solving for the eigenvector corresponding to the smallest eigenvalue of the covariance matrix \mathbf{A} . Thus, a plane can be represented as $\{\mathbf{u}^T(\mathbf{x} - \bar{\mathbf{q}}) = 0 \mid \mathbf{u}, \bar{\mathbf{q}} \in \mathbb{R}^3\}$. To detect whether the foot is in contact with the plane, we define the roughness of the plane by the point-to plane residual:

$$d = \frac{1}{N} \sum_{k=1}^N \mathbf{u}^T (\mathbf{q}_k - \bar{\mathbf{q}}), \quad \mathbf{q}_k \in \mathcal{P}_c. \quad (13)$$

If d is greater than a certain threshold (0.01 m), the plane is considered to have low confidence and is discarded. When the foot is in contact with the plane, the foot is assumed to be coincides with its projection perpendicular to the plane. The foot i 's projection point \mathbf{q}_{proj} can be expressed as:

$$\mathbf{q}_{proj} = {}^W\hat{\mathbf{c}}_i - \mathbf{u}^T({}^W\hat{\mathbf{c}}_i - \bar{\mathbf{q}})\mathbf{u}. \quad (14)$$

Then the contact height measurement can be defined as:

$$\mathbf{e}_z \mathbf{q}_{proj} = \mathbf{h}_{CH}(\mathbf{x}, \mathbf{n}_z) = \mathbf{e}_z^W \mathbf{c}_i + \mathbf{n}_z \quad (15)$$

where $\mathbf{e}_z = [0, 0, 1]$ and the contact height carries a Gaussian noise $\mathbf{n}_z = \mathcal{N}(0, |d\mathbf{e}_z \mathbf{u}|)$, which is correlated with the roughness of the plane.

4) *IMU Measurement*: When the IMU measurements are received, the measurement function of IMU can be expressed:

$$\begin{aligned}\tilde{\mathbf{a}} &= \mathbf{h}_a(\mathbf{x}, \mathbf{n}_{am}) = \mathbf{a} + \mathbf{b}_a + \mathbf{n}_{am} \\ \tilde{\boldsymbol{\omega}} &= \mathbf{h}_\omega(\mathbf{x}, \mathbf{n}_{\omega m}) = \boldsymbol{\omega} + \mathbf{b}_\omega + \mathbf{n}_{\omega m}\end{aligned}\quad (16)$$

where \mathbf{n}_{am} and $\mathbf{n}_{\omega m}$ are Gaussian noised included in the measurement model. Specifically, due to the impact and vibration on the accelerometer measurements, we first examine whether the measurements $\tilde{\mathbf{a}}$ are within the range of measurement rated. If the measurements are saturated, we increase their measurement covariance to a large value.

E. Residual and Update

When the actual measurement $\mathbf{h}(\mathbf{x}, \mathbf{n})$ and the predicted measurement $\mathbf{h}(\hat{\mathbf{x}}, \mathbf{0})$ are known, then the residual \mathbf{r}_{k+1} at step $k+1$ can be linearized as follows:

$$\begin{aligned}\mathbf{r}_{k+1} &= \mathbf{h}(\mathbf{x}_{k+1}, \mathbf{n}_{k+1}) - \mathbf{h}(\hat{\mathbf{x}}_{k+1}, \mathbf{0}) \\ &= \mathbf{h}(\hat{\mathbf{x}}_{k+1} \boxplus \delta \mathbf{x}_{k+1}, \mathbf{n}_{k+1}) - \mathbf{h}(\hat{\mathbf{x}}_{k+1}, \mathbf{0}) \\ &\simeq \mathbf{H}_{k+1} \delta \mathbf{x}_{k+1} + \mathbf{D}_{k+1} \mathbf{n}_{k+1}\end{aligned}\quad (17)$$

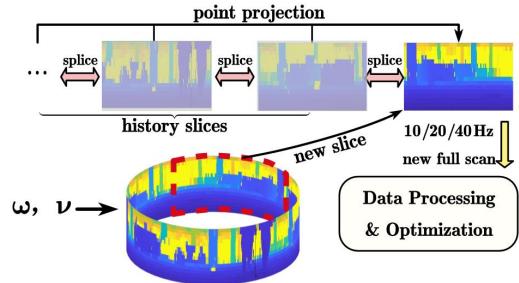


Fig. 5. The overview of Adaptive Scan Slicing and Splicing. A incoming full scan is adaptively partitioned into slices based on the motion velocity estimated by leg odometry. Then, the current slice is spliced with historical slices to obtain a new full scan before ICP.

where \mathbf{H} and \mathbf{D} are the Jacobian matrix of residual \mathbf{r} with respect to \mathbf{x} and \mathbf{n} , respectively. Detailed Jacobian matrices for the various measurements (9),(10),(15),(16) are included in Supplementary ⁵. Finally, the posterior state $\bar{\mathbf{x}}_{k+1}$ can be calculated through the Kalman update process

$$\bar{\mathbf{x}}_{k+1} = \hat{\mathbf{x}}_{k+1} \boxplus \mathbf{K} \mathbf{r}_{k+1} \quad (18)$$

where \mathbf{K} is the Kalman filter gain [22] and \mathbf{R} is the covariance of Gaussian noise \mathbf{n} . And the posterior covariance $\bar{\mathbf{P}}_{k+1}$ can be updated as:

$$\bar{\mathbf{P}}_{k+1} = (\mathbf{I} - \mathbf{K} \mathbf{H}_{k+1}) \hat{\mathbf{P}}_{k+1} \quad (19)$$

IV. LIDAR ODOMETRY

A. Adaptive Scan Slicing and Splicing

Spinning lidar usually takes a certain amount of time (100 ms) to accumulate a 360° FOV full scan, but this process time is too large and causes motion distortion. LLOL [23] and LoLa-SLAM [24] slice the scan according to a specific angle to decrease the latency. They use sliced scan as input instead of full scan, making it possible to get a high rate of lidar odometry.

We propose an adaptive scan slicing and splicing method to alleviate the affects of high-dynamic motion for legged robots. Based on the linear velocity v and angular velocity ω obtained by the leg odometry and converted to the magnitude of vectors in the body frame, the scans are partitioned using an adaptive angle θ_s :

$$\sigma = \left(1 - \max\left(\frac{\omega}{\omega_{\max}}, \frac{v}{v_{\max}}\right)\right) Fov \quad (20)$$

$$\theta_s = \arg \min |m_i - \sigma|, \quad m_i \in [\frac{1}{4}, \frac{1}{2}, 1] Fov \quad (21)$$

where v_{\max} and ω_{\max} are the maximum linear and angular velocity of robot, and Fov is the lidar's 360° FOV. As shown in Fig. 5, a incoming full scan \mathcal{P}_f is partitioned into slices $\{\mathcal{P}_{s1}, \dots, \mathcal{P}_{sn}\}$, which have the size of θ_s . Since slice with smaller FOV will be inaccurate in point cloud matching, we input the historical slices into the buffer, and splice the time-continuous historical slices and current slice into a new

⁵<https://github.com/ouguangjun/Leg-KILO/blob/main/supplementary.pdf>

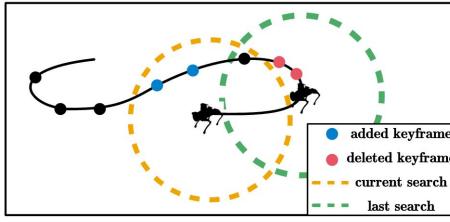


Fig. 6. Robot-centric incremental map. The circular region search is used to determine which lidar keyframes need to be added or deleted from the map. In current search (yellow circle), new keyframes (blue dots) are detected and added to the map, while past keyframes (red dots) are not detected and removed from the map.

full scan \mathcal{P}'_f . In the experiment, we consider that the high-frequency input will aggravate the optimization process, so the output frequency of the new full scan could only be increased to a maximum of 40 Hz (25 ms). Assume that the point q in the historical or current slices in a new full scan is sampled at time t_i . The point q can be projected to the end time t_j of the full scan as: $q' = T_j^{-1}T_i q$, where T_i and T_j are transformations from the body frame to the world frame, which are estimated by leg odometry.

B. Odometry and Mapping

Similar to [17], planes and lines are extracted after receiving the de-skewed scan \mathcal{P}' . According to the initial guess provided by leg odometry, the extracted features will perform point-to-line Φ_l and point-to-plane Φ_p residual calculations with the corresponding features of the local map \mathcal{P}_{map} :

$$\mathbf{r} = \sum_{N_l}^{k=1} \Phi_l(q_{l,k}, \mathcal{P}_{map}) + \sum_{N_p}^{k=1} \Phi_p(q_{p,k}, \mathcal{P}_{map}) \quad (22)$$

where $q_{l,k}, q_{p,k} \in \mathcal{P}'$. After several iterations of the Gauss-Newton method, the residual gradually converges.

In this work, we introduce the lidar keyframe selection concept [3] to reduce the computation of map maintenance. A robot-centric incremental point cloud map is built for the point cloud registration and the contact height detection. As shown in Fig. 6, the circular region centered on the robot is a search area which is used to determine which keyframes need to be updated or deleted from the map. The keyframe will be added to the map or removed as the robot moves. Many works [3], [17] use static kd-tree for the storage of local map and point-wise operations. However, when updating the map, this static data structure needs to be rebuilt for the whole map, which is time-consuming. We employ incremental kd-tree (*ikd-Tree*) [25] to maintain the local map, which can support incrementally adding and deleting point cloud. In the meantime, it also maintains a downsampled map at a given resolution. Multi-thread parallel computation can improve the efficiency of tree re-balancing.

V. FACTOR GRAPH OPTIMIZATION

We utilize factor graphs to tightly couple leg odometry, lidar odometry and loop closure factor. For loop closure, we

follow [3] to apply an efficient distance-based loop closure detection. We choose the state corresponding to the lidar keyframes as the nodes of the factor graph. We define \mathcal{K}_{ij} and \mathcal{L}_{ij} are transformations of leg odometry and lidar odometry, respectively, between node i and node j . And \mathcal{O}_{mn} is defined as the measurement from ICP between two matched nodes m and n . When given the measurements, we can utilize the MAP method to estimate the set of states \mathcal{X}_k^* :

$$\begin{aligned} \mathcal{X}_k^* = \arg \min_{\mathcal{X}_k} & \| \mathbf{r}_0 \|_{\Sigma_0}^2 + \sum_{(m,n) \in C_k} \| \mathbf{r}_{\mathcal{O}_{mn}} \|_{\Sigma_{\mathcal{O}_{mn}}}^2 \\ & + \sum_{(i,j) \in \mathcal{K}_k} (\| \mathbf{r}_{\mathcal{K}_{ij}} \|_{\Sigma_{\mathcal{K}_{ij}}}^2 + \| \mathbf{r}_{\mathcal{L}_{ij}} \|_{\Sigma_{\mathcal{L}_{ij}}}^2) \end{aligned} \quad (23)$$

where \mathbf{r} is the residual between the measurement and the predicted value. Σ is the corresponding covariance.

VI. EXPERIMENTS

We conducted extensive experiments with the Unitree Go1 legged robot both indoors and outdoors. The Unitree Go1 is a highly dynamic robot capable of achieving a maximum speed of 2.0 m/s in our experiences. All experiments are conducted using the trot gait. The equipped with a on-board Velodyne VLP-16 Lidar, as well as a built-in IMU. Each foot of the robot is equipped with three joint encoders to measure joint angle and angular velocity, and a contact sensor for measuring ground contact force. Whether the foot contact is determined by the change of force. Our method is developed based on LIOSAM [3]. Detailed implementation and dataset can be found on our GitHub. The proposed method is tested on a desktop equipped with an Intel i5-12490F processor and 16 GB of memory. Additionally, we also conduct runtime testing on the portable laptop with an Intel i5-7300HQ (2.50 GHz × 4) CPU. At the beginning of each sequence, the robot remains stationary for a period of time to complete state initialization, such as gravity vector, as described in [18].

A. Dataset

Due to the lack of an open source kinematic-inertial-lidar dataset for legged robots, we performed the robot using trot gait, and collected data in serval sequences, containing *corridor*, *parking*, *slope*, *running*, *indoor*, which varying in terms of structure, size, and motion speed. To obtain the ground truth, in sequence *indoor*, we registered the point cloud scan into the prior map by ICP to obtain the trajectory, where the prior map was collected by lidar statically. For other sequences, we employed an offline optimization approach where the previous estimation results were saved and were used as an initial guess for the current ICP algorithm, and the loop detection module was also highly utilized. Multiple iterations were performed until convergence was achieved, allowing us to obtain the high-accuracy results. We also verify methods for end-to-end error on sequence *corridor*, *slope*, and compare the variation of body height while operating on flat ground. More details about experiments are shown in a video <https://youtu.be/6O74De5BLEQ>.

TABLE III
COMPARISON OF RELATIVE POSE ERROR (RPE) FOR OUR PROPOSED METHOD AND OTHER METHODS

Sequences (mean velocity [m/s])	Mean Euler Angle Error [deg] / Translation Error [m]							
	KF [6]	ESKF ⁴ (Ours)	A-LOAM	FAST-LIO2	POINT-LIO	LIO-SAM	LIO-SAM ⁵ (with ESKF)	Ours (w/o LC)
corridor ¹ (0.68)	0.76/0.75	0.95/0.63	0.74/0.18	1.06/0.28	1.00/0.23	× ³	0.68/0.17	0.23/0.08
parking ¹ (0.96)	0.54/1.08	1.11/0.95	0.87/0.23	1.03/0.26	1.09/0.24	×	0.78/0.20	0.17/0.08
indoor ² (0.31)	0.71/0.10	0.54/0.08	0.45/0.04	0.84/0.09	0.72/0.06	0.47/0.04	0.47/0.04	0.15/0.03
running ² (1.50)	0.67/0.17	1.26/0.15	1.08/0.08	2.07/0.19	1.73/0.09	1.10/0.09	1.03/0.10	0.07/0.05

The distance of RPE is set based on the sequence's distance (¹ RPE of 10 m, ² RPE of 1 m). ³ × denotes that the method failed.

⁴ Our proposed kinematic-inertial leg odometry. ⁵ LIO-SAM modified by leg odometry instead of the original IMU odometry.

TABLE IV
COMPARISON OF RELATIVE YAW ANGLE ERROR [DEG]

Method	corridor	parking	indoor	running
KF [6]	0.542	0.468	0.376	0.475
Ours (w/o LC)	0.277	0.215	0.183	0.095

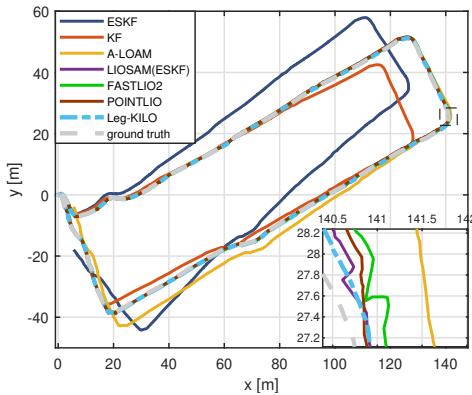


Fig. 7. Comparison of trajectory for various methods on *parking*.

B. Comparison Results

For benchmark results, our proposed method is compared with another kinematic-inertial method (KF [6]), and other state-of-the-art lidar-based methods (A-LOAM [17], LIO-SAM [3], FAST-LIO2 [18], POINT-LIO [19]). KF is a popular state estimator which uses linear Kalman filter to fuse leg kinematics and IMU. For the implementation of KF, we directly use the orientation provided by the 9-axis IMU as the input of the filter. The prediction noise and measurement noise in KF and our proposed leg odometry (ESKF) are the same. Owing to high-frequency impact affecting the IMU accuracy, many lidar-inertial methods encounter significant drift or even breakdown. Consequently, for FAST-LIO2 and POINT-LIO, we increase the covariance of the accelerometer. In the case of LIO-SAM, we turn off the loop detection and additionally replace the original IMU odometry with our proposed leg odometry (LIO-SAM with ESKF). Unless otherwise specified, "Leg-KILO" and "Ours (w/o LC)" refer to versions without loop closure to allow a fair comparison with other algorithms that do not include loop closure. "Ours (full)" represents the version with all components included. We set the map

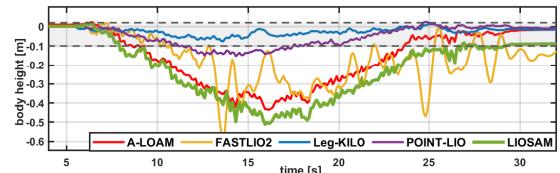


Fig. 8. Comparison of body height's variation for various methods while running on flat ground. The body height will drop slightly when running, and will not exceed the gray area in the figure.

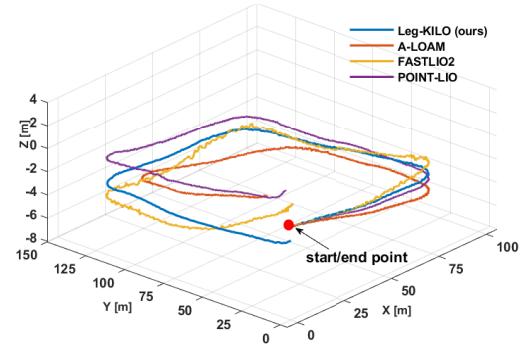


Fig. 9. Comparison of trajectory for various methods on *slope*.

downsampling rate to the same 0.3 for all methods.

1) *Accuracy Comparison*: The comparison of mean Relative Pose Error (RPE) is shown in Table III. It can be seen that our leg odometry ESKF performs best in relative translation compared to another leg odometry KF. This is because ESKF uses the LiDAR map as an additional source of contact height measurements to reduce errors in body height. Due to the lack of observation of the yaw angle, it is difficult to obtain the high-accuracy orientation with leg odometry [4]. We find that the local accuracy of lidar-inertial methods that directly take unstable IMU measurements as input, such as FAST-LIO2 and LIO-SAM, experienced local fluctuations or breakdowns. Incorporating more kinematic constraints, our proposed kinematic-inertial-lidar method Leg-KILO outperforms other methods. For the yaw angle in rotation, the comparison results are shown in Table IV, showing that the yaw angle can be better observed by incorporating an external sensor. The trajectory shown in Fig. 7 and Fig. 9 also validate the high local accuracy and global consistency of Leg-KILO. For comparison of end-to-end error, as shown

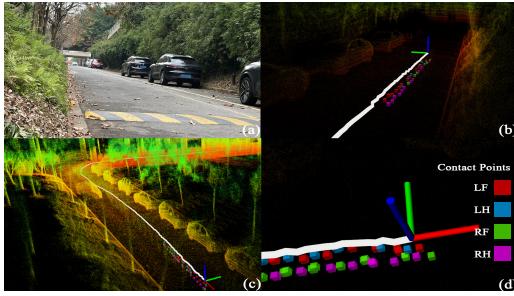


Fig. 10. Experimental results of legged robot on sequence *slope*. (a) The long slope with height variation (exceeding 6 meters). (b) (c) The trajectory with environment mapping results. (d) The foot contact points {LF, LH, RF, RH} measured from Contact Height Detection, which are shown in different colors.

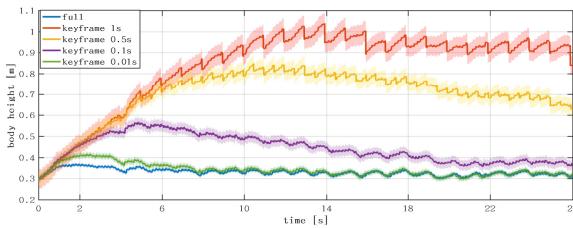


Fig. 11. Comparison of incorporating contact height measurements into the filter at different times on leg odometry in sequence *corridor*. "full" indicates that every measurement is used, while "keyframe i s" indicates that measurements are only added at keyframes, where i represents the time interval between keyframes.

in Table V, many methods obtain drifts less than 0.01% in sequence *corridor*. In sequence *slope*, the robot completed a loop and returned to its initial position in environments with long slopes (height variations exceeding 6 meters), as shown in Fig. 10. Compared to experiments conducted on flat terrain, the performance of our method is more apparent in scenarios with height variations. This is primarily attributed to the integration of leg kinematic constraints to reduce fluctuations in height caused by dynamic motion. The mapping results are shown in Fig. 12.

2) *Body Height's Variation*: In sequence *running*, the robot ran at an average speed of 1.50 m/s on flat ground. The comparison of body height variations among different methods is shown in Fig. 8. Due to the robot's body experiencing slight downward motion during running, we observed height variations within a certain range (gray area in the figure). Several methods exhibit large drift during running and are unable to recover their normal heights during stationary periods. The body height measured by our method can vary within the normal range. Additionally, to investigate the extent to which contact height detection reduces errors in leg odometry, we examine the impact of incorporating contact height measurements into the filter at different stages in sequence *corridor*. The results are shown in Fig. 11. It can be observed that as the keyframe time interval increases (i.e., fewer observations are incorporated), the body height struggles to return to normal levels. This shows the positive effect of contact height detection in mitigating cumulative errors from dynamic movements.

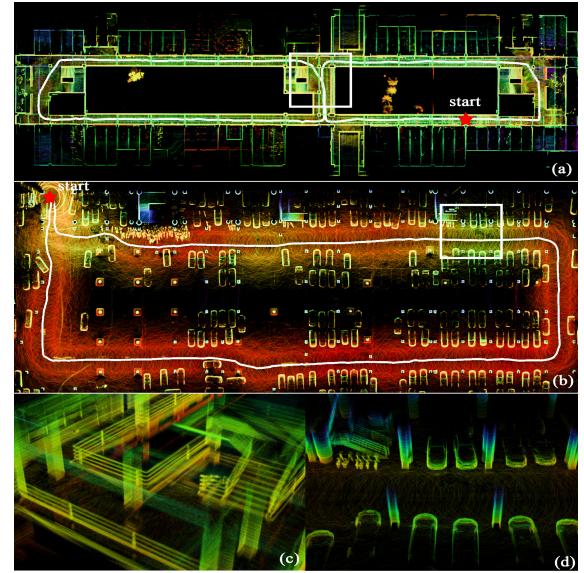


Fig. 12. (a) and (b) show the global mapping results of the sequence *corridor* and *parking*. (c) and (d) are close-up views of the white boxes in (a) and (b), respectively.

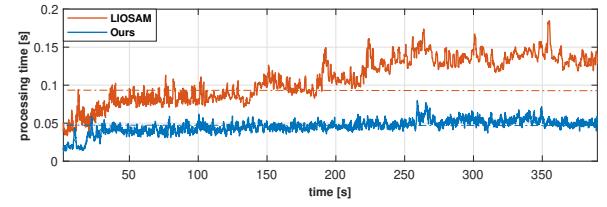


Fig. 13. The runtime comparison on sequence *parking*.

3) *Ablation Study*: As shown in Table III, in LIO-SAM, the IMU odometry obtained by IMU integration exhibits significant drift and even breakdown during long-term operation. Replacing the original IMU odometry with leg odometry ESKF in LIO-SAM has resulted in a notable improvement. From the Table V, it can be observed that the removal of contact height detection (Section III-D3) module has resulted in significant drift, highlighting its critical role. The introduction of loop closure can further reduce global accumulated errors. We simultaneously compare different fixed slicing angles with adaptive slicing angles. From the Table VI, it can be observed that with smaller slicing angles, there is an improvement in accuracy, but CPU usage significantly increases. This is because the ICP thread, which is the primary calculation, becomes more crowded as the number of inputs increases. As the robot's speed increases, the enhancement from slicing slices instead of full scans becomes more pronounced. Adaptive slicing based on the robot's velocity can balance accuracy and computational resources, which is crucial for robots operating with limited computational power.

4) *Efficiency*: The runtime comparison on a portable laptop is shown in Fig. 13. We can find that our method can run at an average rate below 50 ms. The processing time of LIO-SAM increases as the map size grows and exceeds 100 ms. This is because, in LIO-SAM, the map reconstruction during each

TABLE V
COMPARISON OF END-TO-END ERROR [M] FOR VARIOUS METHODS.

	FAST LIO2	POINT LIO	LIO-SAM (with ESKF)	Ours (w/o CHD,LC)	Ours (w/o LC)	Ours (full)
corridor	0.20	0.03	0.10	0.12	0.04	0.03
slope	2.25	3.77	>10	2.34	1.42	0.13

CHD: Contact Height Detection

TABLE VI
ABLATION STUDY OF VARIOUS SLICE SIZES ON SEQUENCE *running*

	360°	180°	90°	Adaptive
ATE [m]	0.063	0.052	0.049	0.055
CPU usage [%]	33.62	50.30	60.81	45.73

update consumes a considerable amount of time, whereas our method allows for incremental updates.

VII. CONSLUSIONS

We propose a Kinematic-Inertial-Lidar Odometry method, Leg-KILO. The integration of leg kinematics constraints enhances the stability of lidar-based odometry. An adaptive scan slicing and splicing method is introduced to alleviate the effect of dynamic motion. Additionally, a robot-centric incremental local mapping approach reduces map maintenance time costs. Factor graph optimization is employed to integrate leg odometry, lidar odometry, and loop closure factors to improve accuracy. Our method is extensively tested in various experiences involving high-dynamic motion using trot gait, demonstrating greater suitability for dynamic quadruped motion compared to other state-of-the-art lidar-based methods. Maintaining robot-centric local maps affects the accuracy of contact height detection. In the future, we will pay more attention to improving the consistency of maps to improve the accuracy of contact height detection and point cloud registration.

REFERENCES

- [1] T. Kim, S. Kim, and D. Lee, "Tunable impact and vibration absorbing neck for robust visual-inertial state estimation for dynamic legged robots," *IEEE Robotics and Automation Letters*, vol. 8, no. 3, pp. 1431–1438, 2023.
- [2] B. R. P. Singh and R. Featherstone, "Mechanical shock propagation reduction in robot legs," *IEEE Robotics and Automation Letters*, vol. 5, no. 2, pp. 1183–1190, 2020.
- [3] T. Shan, B. Englot, D. Meyers, W. Wang, C. Ratti, and D. Rus, "Lio-sam: Tightly-coupled lidar inertial odometry via smoothing and mapping," in *2020 IEEE/RSJ international conference on intelligent robots and systems (IROS)*. IEEE, 2020, pp. 5135–5142.
- [4] M. Bloesch, M. Hutter, M. A. Hoepflinger, S. Leutenegger, C. Gehring, C. D. Remy, and R. Siegwart, "State estimation for legged robots-consistent fusion of leg kinematics and imu," in *Robotics: Science and Systems Conference*, 2012.
- [5] M. Bloesch, C. Gehring, P. Fankhauser, M. Hutter, M. A. Hoepflinger, and R. Siegwart, "State estimation for legged robots on unstable and slippery terrain," in *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2013, pp. 6058–6064.
- [6] G. Bledt, M. J. Powell, B. Katz, J. Di Carlo, P. M. Wensing, and S. Kim, "Mit cheetah 3: Design and control of a robust, dynamic quadruped robot," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2018, pp. 2245–2252.
- [7] R. Hartley, M. Ghaffari, R. M. Eustice, and J. W. Grizzle, "Contact-aided invariant extended kalman filtering for robot state estimation," *The International Journal of Robotics Research*, vol. 39, no. 4, pp. 402–430, 2020.
- [8] J. Ma, M. Bajracharya, S. Susca, L. Matthies, and M. Malchano, "Real-time pose estimation of a dynamic quadruped in gps-denied environments for 24-hour operation," *The International Journal of Robotics Research*, vol. 35, no. 6, pp. 631–653, 2016.
- [9] D. Wisth, M. Camurri, and M. Fallon, "Robust legged robot state estimation using factor graph optimization," *IEEE Robotics and Automation Letters*, vol. 4, no. 4, pp. 4507–4514, 2019.
- [10] S. Yang, Z. Zhang, Z. Fu, and Z. Manchester, "Cerberus: Low-drift visual-inertial-leg odometry for agile locomotion," in *2023 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2023, pp. 4193–4199.
- [11] D. Wisth, M. Camurri, and M. Fallon, "Preintegrated velocity bias estimation to overcome contact nonlinearities in legged robot odometry," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2020, pp. 392–398.
- [12] Y. Kim, B. Yu, E. M. Lee, J.-h. Kim, H.-w. Park, and H. Myung, "Step: State estimator for legged robots using a preintegrated foot velocity factor," *IEEE Robotics and Automation Letters*, vol. 7, no. 2, pp. 4456–4463, 2022.
- [13] V. Dhédin, H. Li, S. Khorshidi, L. Mack, A. K. C. Ravi, A. Meduri, P. Shah, F. Grimminger, L. Righetti, M. Khadiv, et al., "Visual-inertial and leg odometry fusion for dynamic locomotion," in *2023 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2023, pp. 9966–9972.
- [14] D. Wisth, M. Camurri, and M. Fallon, "Vilens: Visual, inertial, lidar, and leg odometry for all-terrain legged robots," *IEEE Transactions on Robotics*, vol. 39, no. 1, pp. 309–326, 2022.
- [15] J.-H. Kim, S. Hong, G. Ji, S. Jeon, J. Hwangbo, J.-H. Oh, and H.-W. Park, "Legged robot state estimation with dynamic contact event information," *IEEE Robotics and Automation Letters*, vol. 6, no. 4, pp. 6733–6740, 2021.
- [16] R. Hartley, M. G. Jadidi, L. Gan, J.-K. Huang, J. W. Grizzle, and R. M. Eustice, "Hybrid contact preintegration for visual-inertial-contact state estimation using factor graphs," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2018, pp. 3783–3790.
- [17] J. Zhang and S. Singh, "Loam: Lidar odometry and mapping in real-time," in *Robotics: Science and systems*, vol. 2, no. 9. Berkeley, CA, 2014, pp. 1–9.
- [18] W. Xu, Y. Cai, D. He, J. Lin, and F. Zhang, "Fast-lio2: Fast direct lidar-inertial odometry," *IEEE Transactions on Robotics*, vol. 38, no. 4, pp. 2053–2073, 2022.
- [19] D. He, W. Xu, N. Chen, F. Kong, C. Yuan, and F. Zhang, "Point-lio: Robust high-bandwidth light detection and ranging inertial odometry," *Advanced Intelligent Systems*, p. 2200459, 2023.
- [20] M. Tranzatto, M. Dharmadhikari, L. Bernreiter, M. Camurri, S. Khattak, F. Mascalich, P. Pfreundschuh, D. Wisth, S. Zimmermann, M. Kulkarni, et al., "Team cerberus wins the darpa subterranean challenge: Technical overview and lessons learned," *arXiv preprint arXiv:2207.04914*, 2022.
- [21] B. Morrell, R. Thakker, Á. Santamaría Navarro, A. Bouman, X. Lei, J. Edlund, T. Pailevanian, T. S. Vaquero, Y. L. Chang, T. Touma, et al., "Nebula: Team costar's robotic autonomy solution that won phase ii of darpa subterranean challenge," *Field robotics*, vol. 2, pp. 1432–1506, 2022.
- [22] P. S. Maybeck, *Stochastic models, estimation, and control*. Academic press, 1982.
- [23] C. Qu, S. S. Shivakumar, W. Liu, and C. J. Taylor, "Llol: Low-latency odometry for spinning lidars," in *2022 International Conference on Robotics and Automation (ICRA)*. IEEE, 2022, pp. 4149–4155.
- [24] M. Karimi, M. Oelsch, O. Stengel, E. Babaian, and E. Steinbach, "Lola-slam: Low-latency lidar slam using continuous scan slicing," *IEEE Robotics and Automation Letters*, vol. 6, no. 2, pp. 2248–2255, 2021.
- [25] Y. Cai, W. Xu, and F. Zhang, "ikd-tree: An incremental kd tree for robotic applications," *arXiv preprint arXiv:2102.10808*, 2021.