2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)
October 14-18, 2024. Abu Dhabi, UAE

KLILO: Kalman Filter based LiDAR-Inertial-Leg Odometry for Legged Robots

Shaohang Xu[1,2], Wentao Zhang[1], and Lijun Zhu[1]

***Abstract*— This paper presents a Kalman filter based LiDAR-Inertial-Leg Odometry (KLILO) system for legged robots to navigate in challenging environments. In particular, we employ the iterated error-state extended Kalman filter framework on manifolds to fuse measurements from the inertial measurement unit (IMU), LiDAR, joint encoders, and contact force sensors in a tightly coupled manner. To assess the performance of KLILO, we build a dataset that encompasses intricate environments with challenging conditions such as dynamic objects and deformable terrains. The results demonstrate that our algorithm can provide efficient and reliable localization in all tests. It exhibits an average improvement of around 40% in positioning accuracy compared to the baselines. Furthermore, we validate KLILO in a challenging navigation task on a real robot, where the LiDAR encounters ineffective measurements.**

I. Introduction

State estimation is critical for the practical deployment of legged robots in real-world environments, as these robots depend on precise localization capabilities to plan and execute optimal trajectories. The primary state estimators for legged robots are based on proprioceptive sensors, combining legged kinematics and IMU data through Kalman filtering [1]–[3]. However, these proprioceptive methods can suffer performance degradation due to deformable terrains and model inaccuracies. In particular, this issue becomes notably exacerbated during dynamic locomotion.

On the other hand, exteroceptive alternatives, such as LiDAR-inertial odometry (LIO) [4], [5] and visual-inertial odometry (VIO) [6], are less influenced by locomotion kinematics. Nevertheless, these exteroceptive solutions also face challenges, including scenarios with low lighting, moving obstacles, expansive surfaces, and more.

In this paper, we aim to propose a robust and effective state estimation algorithm for real-world applications of legged robots. To this end, we take advantage of both proprioceptive and exteroceptive sensors to improve the robustness.

Fig. 1 shows a demonstration scenario where the robot encounters an obstructing shade cloth during its operation, causing the camera or LiDAR data to become less useful and making exteroceptive-only methods ineffective. Similarly, as shown in Fig. 2, proprioceptive state estimators can quickly drift due to dynamic locomotion and complex terrains.

To address these challenges, we propose to use the iterated error-state extended Kalman filter on manifolds [7] and seamlessly integrate the measurements of IMU, LiDAR, and legged kinematics in a tightly coupled manner. To the best of our knowledge, this is the first tightly coupled, Kalman filter based LiDAR-Inertial-Leg odometry system in the literature. This paper makes the following specific contributions:

- We propose a compact LiDAR-Inertial-Leg odometry system, KLILO, for legged robots. In particular, the measurements of IMU, LiDAR, and legged kinematics are tightly coupled in an iterated error-state extended Kalman filtering framework.
- We conduct extensive data collection using a quadruped robot platform equipped with a diverse range of sensors, which possess distinct and formidable characteristics relevant to legged locomotion.
- Through extensive experiments, our algorithm demonstrates superior accuracy compared to proprioceptive-only or exteroceptive-only algorithms. Moreover, it is validated online in a challenging navigation task.

The remainder of this paper is organized as follows. Section

Fig. 1. The navigation application with KLILO. During the navigation, the shade cloth is put onto the robot, such that the LiDAR points become ineffective.

Fig. 2. The test scenarios in the experiments. The subfigures (a)-(d) show the scenarios in Test 1-4, respectively.

[1] School of Artificial Intelligence and Automation, Huazhong University of Science and Technology, China, shaohangxu@hust.edu.cn, wentaozhang@hust.edu.cn, ljzhu@hust.edu.cn

[2] School of Data Science, City University of Hong Kong, HKSAR, shaohanxu2-c@my.cityu.edu.hk

* The code and data will be available [8].

II provides a brief review of related work on state estimation in legged robots. In Section III, we present our proposed odometry system in detail, including the measurement models and the Kalman filter algorithm. In Section IV, we show the experimental results. Finally, Section V concludes this paper.

II. Related Work

Early state estimation algorithms for legged robots were primarily focused on proprioceptive algorithms, involving measurements of IMUs, joint encoders, and contact force sensors. In particular, most of these algorithms relied on Kalman filter techniques [1]–[3]. In [1], an extended Kalman filter is proposed to integrate legged forward kinematics with a single IMU. To tackle challenges caused by unstable and slippery terrains, an outlier detection mechanism is further proposed in [2] based on the Mahalanobis distance. Hartley et al. [3] developed an invariant extended Kalman filter to improve the orientation estimation accuracy, which is based on Lie group theory and invariant observer design principles.

While proprioceptive methods excel in high-frequency feedback control scenarios, their performance tends to degrade in long-distance navigation applications due to their drift errors. In particular, the drift error can theoretically grow without bound over time, which has been proven in [1]. This necessitates the incorporation of exteroceptive sensors.

Among exteroceptive sensors, vision sensors have emerged as prime candidates due to their lightweight and cost-effective nature [9]–[13]. For instance, Hartley et al. [9] proposed a hybrid contact preintegration method based on factor graphs, which integrates contact data with camera information. This approach helps to reduce the complexity of nonlinear optimization problems and enhances estimation accuracy while improving robustness to visual disturbances. Wisth et al. [10] proposed directly integrating visual features into the cost function along with preintegrated factors, which showed promising results on uneven and slippery terrains. However, vision sensors suffer from low lighting and motion blur. Moreover, real-time dense mapping is challenging for vision-based algorithms, particularly due to the limited computation resources on the real robot.

LiDAR, with its ability to provide direct, dense, active, and accurate depth measurements, is a powerful sensor in the context of state estimation for legged robots. A pioneering work is [14], where the LiDAR-derived factors were integrated into a factor graph optimization framework with visual and legged measurements. This work demonstrated impressive results in field experiments such as the DARPA Subterranean Challenge, proving particularly beneficial for dense mapping and perceptive locomotion. However, it is noteworthy that most exteroceptive algorithms for legged robots focus on smoothing, which is not suitable for high-frequency feedback planning and control due to the computation time. In contrast, Kalman filtering is computationally efficient on legged robots, but remains unexplored in the context of LiDAR-Inertial-Leg odometry.

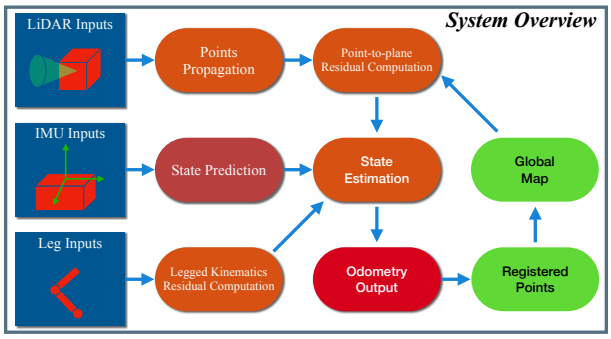

Fig. 3. System overview of KLILO.

III. The KLILO Algorithm

A. System Overview

The overview of our algorithm is shown in Fig. 3. Initially, the IMU measurements are used to compute the predicted states, and the first IMU frame serves as the global frame. Using measurements of joint encoders and contact sensors, the residual term for the legged measurement model can be calculated. Subsequently, LiDAR points are accumulated over a user-defined time period (e.g., 100 milliseconds in our experiments) to form a scan. Due to motion distortion, the raw scan is compensated through points propagation using the predicted states. This process corrects any distortions caused by the motion of the robot during the LiDAR scanning period, which is particularly important for ensuring the accuracy of the subsequent steps. For each point in the scan, we compute the related point-to-plane residual based on the global map. The above residual terms are tightly fused with the IMU predicted states in an iterated Kalman filter framework. Once the iterative Kalman filter has converged, the final estimated states are obtained, which will be used to register the LiDAR points into the global map.

B. IMU Kinematics Model and State Prediction

We first define the operators $\oplus$ and $\ominus$ as follows:

$$\begin{bmatrix} \mathbf{R} \\ \mathbf{a} \end{bmatrix} \oplus \begin{bmatrix} \mathbf{r} \\ \mathbf{b} \end{bmatrix} \triangleq \begin{bmatrix} \mathbf{R} \cdot \mathrm{Exp}(\mathbf{r}) \\ \mathbf{a}+\mathbf{b} \end{bmatrix},$$

$$\begin{bmatrix} \mathbf{R}_2 \\ \mathbf{a} \end{bmatrix} \ominus \begin{bmatrix} \mathbf{R}_1 \\ \mathbf{b} \end{bmatrix} \triangleq \begin{bmatrix} \mathrm{Log}(\mathbf{R}_1^\top \mathbf{R}_2) \\ \mathbf{a}-\mathbf{b} \end{bmatrix},$$

where $\mathbf{R}, \mathbf{R}_1, \mathbf{R}_2 \in SO(3)$, $\mathbf{r} \in \mathbb{R}^3$, $\mathbf{a}, \mathbf{b} \in \mathbb{R}^{15}$, $\mathrm{Exp}(\cdot)$ is the exponential map on $SO(3)$ and $\mathrm{Log}(\cdot)$ is its inverse map. Define $\mathcal{M} \triangleq SO(3) \times \mathbb{R}^{15}$. Then, the IMU kinematics model is:

$$\mathbf{x}_{i+1} = \mathbf{x}_i \oplus (\mathbf{f}(\mathbf{x}_i, \mathbf{u}_i, \mathbf{w}_i)\Delta t),$$

where the subscript i represents the item at the time step i. The state $\mathbf{x}$, the dynamics function $\mathbf{f}$, the control input $\mathbf{u}$ and the process noise $\mathbf{w}$ are defined as

$$\mathbf{x} \in \mathcal{M} \triangleq [{}^G_I\mathbf{R}^\top \quad {}^G\mathbf{p}^\top \quad {}^G\mathbf{v}^\top \quad \mathbf{b}^{\omega\top} \quad \mathbf{b}^{a\top} \quad {}^G\mathbf{g}^\top]^\top,$$

$$\mathbf{u} \triangleq [\boldsymbol{\omega}_m^\top \quad \mathbf{a}_m^\top]^\top, \quad \boldsymbol{w} \triangleq [\mathbf{n}^{\omega\top} \quad \mathbf{n}^{a\top} \quad \mathbf{n}^{b\omega\top} \quad \mathbf{n}^{ba\top}]^\top,$$

$$\mathbf{f}(\mathbf{x},\mathbf{u},\mathbf{w}) \triangleq \begin{bmatrix} \boldsymbol{\omega}_m - \mathbf{b}^{\omega} - \mathbf{n}^{\omega} \\ {}^{G}\mathbf{v} + \frac{1}{2}({}^{G}_{I}\mathbf{R}(\mathbf{a}_m - \mathbf{b}^{a} - \mathbf{n}^{a}) + {}^{G}\mathbf{g})\Delta t \\ {}^{G}_{I}\mathbf{R}(\mathbf{a}_m - \mathbf{b}^{a} - \mathbf{n}^{a}) + {}^{G}\mathbf{g} \\ \mathbf{n}^{b\omega} \\ \mathbf{n}^{ba} \\ \mathbf{0}_{3\times 1} \end{bmatrix},$$

where I represents the IMU frame, G represents the global frame, ${}^{G}_{I}\mathbf{R}$ and ${}^{G}\mathbf{p}$ denote the IMU attitude and position in the global frame, ${}^{G}\mathbf{v}$ is the IMU linear velocity in the global frame, $\mathbf{b}^{\omega}$ and $\mathbf{b}^{a}$ are the IMU biases, ${}^{G}\mathbf{g}$ is the gravity vector in the global frame, $\boldsymbol{\omega}_m$ and $\mathbf{a}_m$ are the raw IMU measurements, $\mathbf{n}^{\omega}, \mathbf{n}^{a}, \mathbf{n}^{b\omega}$ and $\mathbf{n}^{ba}$ are all Gaussian noises. Given the optimal state estimation $\bar{\mathbf{x}}_k$ at the time step k, the predictive state at the time step $i+1$ ($i \geq k$) is then calculated as

$$\hat{\mathbf{x}}_{i+1} = \hat{\mathbf{x}}_i \oplus (\mathbf{f}(\hat{\mathbf{x}}_i, \mathbf{u}_i, \mathbf{0})\Delta t), \ \hat{\mathbf{x}}_k = \bar{\mathbf{x}}_k, \tag{1}$$

where $\hat{\mathbf{x}}_i$ denotes the predictive state at the time step i, and Δt denotes the time window between two inertial measurements.

The predictive covariance is propagated as follows. First, we define the error state as

$$\delta\hat{\mathbf{x}}_i \triangleq \mathbf{x}_i \ominus \hat{\mathbf{x}}_i.$$

Then, we rewrite the error dynamics $\delta\hat{\mathbf{x}}_i$ using the IMU kinematics and the predictive state:

$$\begin{aligned} \delta\hat{\mathbf{x}}_i &= \mathbf{x}_i \ominus \hat{\mathbf{x}}_i \\ &= (\mathbf{x}_{i-1} \oplus (\mathbf{f}(\mathbf{x}_{i-1}, \mathbf{u}_{i-1}, \mathbf{w}_{i-1})\Delta t)) \\ &\quad \ominus (\hat{\mathbf{x}}_{i-1} \oplus (\mathbf{f}(\hat{\mathbf{x}}_{i-1}, \mathbf{u}_{i-1}, \mathbf{0})\Delta t)) \\ &\approx \mathbf{F}_{\delta\hat{\mathbf{x}}_{i-1}}\delta\hat{\mathbf{x}}_{i-1} + \mathbf{F}_{\mathbf{w}_{i-1}}\mathbf{w}_{i-1}. \end{aligned}$$

Note that the approximation terms $\mathbf{F}_{\delta\hat{\mathbf{x}}_i}$ and $\mathbf{F}_{\mathbf{w}_i}$ are computed by the first-order Taylor expansion.

Denote $\mathbf{Q}_i$ as the covariance matrix of the noise $\mathbf{w}_i$. Then, the covariance $\hat{\mathbf{P}}_i$ related to the predictive estimation $\hat{\mathbf{x}}_i$ is

$$\hat{\mathbf{P}}_{i+1} = \mathbf{F}_{\delta\hat{\mathbf{x}}_i}\hat{\mathbf{P}}_i\mathbf{F}^{\top}_{\delta\hat{\mathbf{x}}_i} + \mathbf{F}_{\mathbf{w}_i}\mathbf{Q}_i\mathbf{F}^{\top}_{\mathbf{w}_i}, \hat{\mathbf{P}}_k = \bar{\mathbf{P}}_k, \tag{2}$$

where $\bar{\mathbf{P}}_k$ denotes the covariance related to the optimal estimation $\bar{\mathbf{x}}_k$ at the time step k.

C. LiDAR Measurement Model

LiDAR typically samples points sequentially, and thus the raw points are sampled at different poses as the LiDAR moves. The data acquired from the IMU often has a higher sampling rate compared to LiDAR data. Therefore, we employ the predictive state $\hat{x}_i$ (1) to project all raw points to the end time of the scan. Consequently, the points within the scan can be treated as if they were simultaneously sampled at the end time of the scan.

Let us denote ${}^{L}\mathbf{l}^{\alpha}_{k}$ as the position of the α-th LiDAR point in the LiDAR frame L at time step k. In order to transform ${}^{L}\mathbf{l}^{\alpha}_{k}$ into the global frame G, we can utilize the transformation from the body frame to the global frame: ${}^{G}_{I}\mathbf{T} \triangleq ({}^{G}_{I}\mathbf{R}, \ {}^{G}\mathbf{p})$, where ${}^{G}_{I}\mathbf{R}$ represents the rotation matrix and ${}^{G}\mathbf{p}$ denotes the translation vector. Then, we have the following equation to calculate the α-th LiDAR point in the global frame at the time step k:

$${}^{G}\mathbf{l}^{\alpha}_{k} = {}^{G}_{I}\mathbf{R}\ ({}^{I}_{L}\mathbf{R}\ {}^{L}\mathbf{l}^{\alpha}_{k} + {}^{I}_{L}\mathbf{p}) + {}^{G}\mathbf{p}.$$

Here, ${}^{I}_{L}\mathbf{T} \triangleq ({}^{I}_{L}\mathbf{R}, \ {}^{I}_{L}\mathbf{p})$ refers to the extrinsic matrix that describes the transformation between the LiDAR frame and the IMU frame. Utilizing the map established by the ikd-tree algorithm [4], we search and identify the five closest points to ${}^{G}\mathbf{l}^{\alpha}_{k}$ within the global map. Then, we fit a plane to these points with the normal vector denoted as $\mathbf{n}^{\alpha}_{k}$ and the center point of the plane denoted as ${}^{G}\mathbf{c}^{\alpha}_{k}$. In an ideal scenario, the LiDAR point ${}^{G}\mathbf{l}^{\alpha}_{k}$ should precisely lie within the fitted plane, leading to the LiDAR measurement model:

$$\begin{aligned} h^{L}_{\alpha}(\mathbf{x}_k) \triangleq \mathbf{n}^{\alpha\top}_{k}\left({}^{G}\mathbf{l}^{\alpha}_{k} - {}^{G}\mathbf{c}^{\alpha}_{k}\right) = 0, \\ \forall \alpha \in \{1, \cdots, N^{L}_{k}\}, \end{aligned} \tag{3}$$

where N^{L}_{k} denotes the quantity of the LiDAR points in the time step k. As can be seen, we use all the points in the LiDAR scan without extracting any features.

D. Leg Measurement Model

Denote $\boldsymbol{\phi}^{\beta}_{k}$ as the vector of the joint angles of the β-th contact leg at the time step k and $\dot{\boldsymbol{\phi}}^{\beta}_{k}$ as the corresponding joint velocities. Based on the forward kinematics, we can obtain the foot position in the robot body frame:

$${}^{I}\mathbf{q}^{\beta}_{k} = \boldsymbol{\Psi}^{\beta}(\boldsymbol{\phi}^{\beta}_{k}),$$

where ${}^{I}\mathbf{q}^{\beta}_{k}$ is the β-th foot position in the body frame I at the time step k, $\boldsymbol{\Psi}^{\beta}(\cdot)$ is the forward kinematics map for the β-th leg. Then, in the global frame, we have

$${}^{G}\mathbf{q}^{\beta}_{k} = {}^{G}\mathbf{p} + {}^{G}_{I}\mathbf{R}\ {}^{I}\mathbf{q}^{\beta}_{k},$$

where ${}^{G}\mathbf{q}^{\beta}_{k}$ is the corresponding position in the global frame. In this work, the contact state could be identified by a force sensor on each robot foot. Then, if the β-th leg is a stance leg, there should be no slippage between the foot and the ground, i.e.,

$$\begin{aligned} h^{Q}_{\beta}(\mathbf{x}) &\triangleq {}^{G}\mathbf{v} + {}^{G}_{I}\dot{\mathbf{R}}\boldsymbol{\Psi}^{\beta}(\boldsymbol{\phi}^{\beta}) + {}^{G}_{I}\mathbf{R}\tfrac{\mathrm{d}}{\mathrm{d}t}\boldsymbol{\Psi}^{\beta}(\boldsymbol{\phi}^{\beta}) \\ &= {}^{G}\mathbf{v} + {}^{G}_{I}\mathbf{R}\lfloor\boldsymbol{\omega}_m\rfloor^{\wedge}\boldsymbol{\Psi}^{\beta}(\boldsymbol{\phi}^{\beta}) + {}^{G}_{I}\mathbf{R}\mathbf{J}^{\beta}(\boldsymbol{\phi}^{\beta})\dot{\boldsymbol{\phi}}^{\beta} \\ &= 0, \\ &\beta \in \{1, \cdots, N^{Q}_{k}\}, \end{aligned} \tag{4}$$

where N^{Q}_{k} denotes the number of contact legs, $\lfloor\cdot\rfloor^{\wedge}$ is the skew-symmetric operator, $\mathbf{J}^{\beta}(\cdot)$ is the Jacobian matrix of the β-th leg.

E. Iterated Error-state Extended Kalman Filter Update

Given the predictive state, we have the prior distribution $\mathbf{x}_k \ominus \hat{\mathbf{x}}_k \sim \mathcal{N}(0, \hat{\mathbf{P}}_k)$. Then, combining the measurement models (3) and (4), we have the following maximum a posteriori estimation problem:

$$\begin{aligned} \min_{\mathbf{x}_k} \ & \Big(\|\mathbf{x}_k \ominus \hat{\mathbf{x}}_k\|^2_{\hat{\mathbf{P}}^{-1}_k} + \textstyle\sum_{\alpha} \|h^{L}_{\alpha}(\mathbf{x}_k)\|^2_{\boldsymbol{\Sigma}_L} \\ & + \textstyle\sum_{\beta} \|h^{Q}_{\beta}(\mathbf{x}_k)\|^2_{\boldsymbol{\Sigma}_Q} \Big), \end{aligned}$$

where $\|\mathbf{x}\|^2_{\mathbf{\Sigma}} \triangleq \mathbf{x}^\top \mathbf{\Sigma} \mathbf{x}$. To address this issue, we leverage the iterated error-state extended Kalman filter [7] to deal with the manifolds.

First, we reformulate the LiDAR residual $h^L_\alpha(\mathbf{x}_k)$ as follows. Denote $\hat{\mathbf{x}}^\kappa_k$ as the update state at the κ-th iteration, and define $\hat{\mathbf{x}}^0_k \triangleq \hat{\mathbf{x}}_k$, i.e., the predictive state. Then, at the $(\kappa+1)$-th iteration, we take the first-order Taylor approximation as

$$h^L_\alpha(\mathbf{x}_k) \approx h^L_\alpha(\hat{\mathbf{x}}^\kappa_k) + \partial h^L_\alpha(\hat{\mathbf{x}}^\kappa_k)(\mathbf{x}_k \ominus \hat{\mathbf{x}}^\kappa_k),$$

where $\partial h^L_\alpha(\hat{\mathbf{x}}_k)$ is the Jacobian matrix of $h^L_\alpha(\cdot)$. From the definition (3), we know that $h^L_\alpha(\cdot)$ is only dependent on ${}^G\mathbf{p}$ and ${}^G_I\mathbf{R}$. For ${}^G\mathbf{p}$, the related Jacobian is $\mathrm{diag}\{\mathbf{n}^\alpha_k\}$, while for ${}^G_I\mathbf{R}$, it is $\mathrm{diag}\{-\lfloor {}^G_I\mathbf{R}({}^I_L\mathbf{R}\ {}^L\mathbf{l}^\alpha_k + {}^I_L\mathbf{p})\rfloor^\wedge \mathbf{n}^\alpha_k\}$. Then we have $\partial h^L_\alpha(\mathbf{x})$ as:

$$\mathrm{diag}\{-\lfloor {}^G_I\mathbf{R}({}^I_L\mathbf{R}\ {}^L\mathbf{l}^\alpha_k + {}^I_L\mathbf{p})\rfloor^\wedge \mathbf{n}^\alpha_k, \mathbf{n}^\alpha_k, \mathbf{0}, \mathbf{0}, \mathbf{0}, \mathbf{0}\}.$$

Similarly, the legged residual term $h^Q_\beta(\mathbf{x}_k)$ can be linearized as follows:

$$h^Q_\beta(\mathbf{x}_k) \approx h^Q_\beta(\hat{\mathbf{x}}^\kappa_k) + \partial h^Q_\beta(\hat{\mathbf{x}}^\kappa_k)(\mathbf{x}_k \ominus \hat{\mathbf{x}}^\kappa_k),$$

where $\partial h^Q_\beta(\cdot)$ is the Jacobian matrix of $h^Q_\beta(\cdot)$. For ${}^G\mathbf{v}$, the related Jacobian is clearly $\mathbf{I}_{3\times3}$; For ${}^G_I\mathbf{R}$, it is $-\lfloor {}^G_I\mathbf{R}(\lfloor \boldsymbol{\omega}_m \rfloor^\wedge \mathbf{\Psi}(\boldsymbol{\phi}) + \mathbf{J}^\beta(\boldsymbol{\phi}^\beta)\dot{\boldsymbol{\phi}}^\beta)\rfloor^\wedge$. Then we can summarize that $\partial h^Q_\beta(\mathbf{x})$ is

$$\mathrm{diag}\{-\lfloor {}^G_I\mathbf{R}(\lfloor \boldsymbol{\omega}_m \rfloor^\wedge \mathbf{\Psi}(\boldsymbol{\phi}) + \mathbf{J}^\beta(\boldsymbol{\phi}^\beta)\dot{\boldsymbol{\phi}}^\beta)\rfloor^\wedge, \mathbf{0}, \mathbf{I}_{3\times3}, \mathbf{0}, \mathbf{0}, \mathbf{0}\}.$$

Given the linearized residual terms, we can transform the original problem into the following:

$$\begin{aligned}\min_{\mathbf{x}_k} \quad & \Big(\|\mathbf{x}_k \ominus \hat{\mathbf{x}}_k\|^2_{\hat{\mathbf{P}}_k^{-1}} \\ & + \textstyle\sum_\alpha \|h^Q_\alpha(\hat{\mathbf{x}}^\kappa_k) + \partial h^Q_\beta(\hat{\mathbf{x}}^\kappa_k)(\mathbf{x}_k \ominus \hat{\mathbf{x}}^\kappa_k)\|^2_{\mathbf{\Sigma}_L} \\ & + \textstyle\sum_\beta \|h^Q_\beta(\hat{\mathbf{x}}^\kappa_k) + \partial h^Q_\beta(\hat{\mathbf{x}}^\kappa_k)(\mathbf{x}_k \ominus \hat{\mathbf{x}}^\kappa_k)\|^2_{\mathbf{\Sigma}_Q} \Big),\end{aligned}$$

The above optimization problem could be solved by the Gaussian-Newton method, which is equivalent to the iterated Kalman filter [15]. The Kalman gain matrix is

$$\mathbf{K} = \mathbf{P}\mathbf{H}^\top(\mathbf{H}\mathbf{P}\mathbf{H}^\top + \mathbf{\Sigma})^{-1},$$

where

$$\mathbf{H} \triangleq [\partial h^L_1(\mathbf{x}), \cdots, \partial h^Q_{N^L_k}(\mathbf{x}), \partial h^Q_1(\mathbf{x}), \cdots, \partial h^Q_{N^Q_k}(\mathbf{x})],$$

$$\mathbf{\Sigma} \triangleq \mathrm{diag}\{\mathbf{\Sigma}_L, \cdots, \mathbf{\Sigma}_L, \mathbf{\Sigma}_Q, \cdots, \mathbf{\Sigma}_Q\},$$

$$\mathbf{P} \triangleq (\mathbf{J}^\kappa)^{-1}\hat{\mathbf{P}}_k\mathbf{J}^\kappa,$$

$\mathbf{J}^\kappa$ is the partial differentiation of $\hat{\mathbf{x}}^\kappa_k \oplus \delta\mathbf{x}^\kappa_k \ominus \hat{x}_k$ w.r.t $\delta\mathbf{x}^\kappa_k$:

$$\mathbf{J}^\kappa \triangleq \begin{bmatrix} \mathbf{A}({}^G_{I_k}\hat{\mathbf{R}}^\kappa \ominus {}^G_{I_k}\hat{\mathbf{R}})^{-\top} & \mathbf{0} \\ \mathbf{0} & \mathbf{I}_{15\times15} \end{bmatrix},$$

where the operator $\mathbf{A}(\cdot)^{-1}$ is defined as

$$\mathbf{A}(x) = \mathbf{I} - \frac{1}{2}\lfloor \mathbf{x} \rfloor^\wedge + (1 - \frac{\|\mathbf{x}\|}{2}\cot(\frac{\|\mathbf{x}\|}{2})\frac{(\lfloor \mathbf{x} \rfloor^\wedge)^2}{\|\mathbf{x}\|^2}.$$

We can update the state estimation based on the Kalman gain as follows [4]:

$$\hat{\mathbf{x}}^{\kappa+1}_k = \hat{\mathbf{x}}^\kappa_k \oplus (-\mathbf{K}\mathbf{h}^\kappa_k - (\mathbf{I} - \mathbf{K}\mathbf{H})(\mathbf{J}^\kappa)^{-1}(\hat{\mathbf{x}}^\kappa_k \ominus \hat{\mathbf{x}}_k)),$$

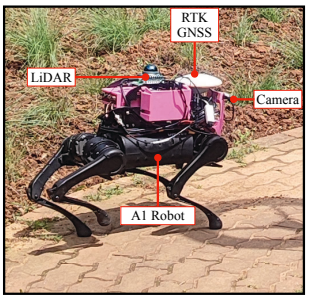

Fig. 4. The A1 robot with the sensor suite. In the outdoor environments, we use RTK GNSS to obtain the ground truth. The camera in this paper is only used for recording.

where

$$\mathbf{h}^\kappa_k \triangleq [h^L_1(\hat{\mathbf{x}}^\kappa_k), \cdots, h^Q_{N^L_k}(\hat{\mathbf{x}}^\kappa_k), h^Q_1(\hat{\mathbf{x}}^\kappa_k), \cdots, h^Q_{N^Q_k}(\hat{\mathbf{x}}^\kappa_k)].$$

The convergence condition is $\|\hat{\mathbf{x}}^{\kappa-1}_k - \hat{\mathbf{x}}^\kappa_k\| \le \epsilon$, where ϵ is the tolerance. The update state at the final iteration step κ is defined as the optimal estimation, i.e., $\bar{\mathbf{x}}_k \triangleq \hat{\mathbf{x}}^\kappa_k$, and the covariance

$$\bar{\mathbf{P}}_k \triangleq (\mathbf{I} - \mathbf{K}\mathbf{H})\mathbf{P}.$$

Then, $\bar{\mathbf{x}}$ and $\bar{\mathbf{P}}_k$ will be again utilized in the state prediction process to propagate the future IMU measurements according to Equation (1) and (2). Additionally, the optimal estimated states are used to update the global maps by registering the LiDAR points. Note that the measurements of LiDAR and legged kinematics are not guaranteed to be perfectly synchronized. Hence, if only a LiDAR scan is received at time step k, the state estimation will be updated solely based on the LiDAR residual term. The same applies to the legged kinematics measurement.

IV. Experimental Results

A. Hardware

We conduct data collection on a Unitree A1 robot [16] equipped with a customized sensor suite, as shown in Fig. 4. This robot has 12 high-performance servo motors with joint encoders. Moreover, there are force sensors at each robot foot. The readings from these force sensors are directly utilized to determine the leg states. The data from joint and contact sensors enable the calculation of the legged kinematics, as elaborated in Section III-D. The A1 robot includes a six-axis IMU with finely tuned parameters. A hard synchronization has been enforced between the IMU, joint, and contact sensors.

A Livox Mid-360 LiDAR has been installed on the robot back. A non-repetitive scanning mode is used to output the point cloud data at a frequency of 10 [Hz]. However, it is worth noting that the data from the Mid-360 cannot be hard synchronized with the IMU data. Instead, the LI_init algorithm [17] is utilized for calibration, encompassing the transform matrix and offset time.

The above-mentioned sensors supply all the necessary input data for the KLILO algorithm. Additionally, an Intel

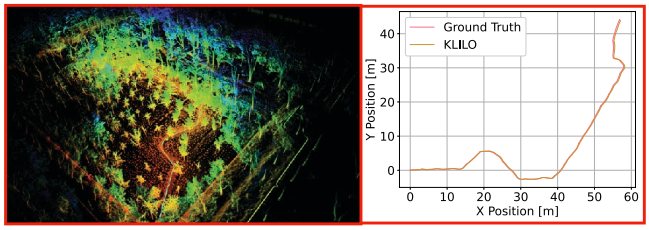

Fig. 5. Qualitative Results of Test 2. (left) The dense map constructed in real-time. The computed trajectory is colored in blue. (right) The RTK GNSS trajectory (ground truth) and the KLILO trajectory.

TABLE I
QUANTITIVE RESULTS (10M RPE)

Data	LEO	Fast-LIO2	Ours
Test 1	0.80(0.23)	0.31(0.02)	**0.27**(0.02)
Test 2	2.26(1.90)	0.66(0.16)	**0.49**(0.14)
Test 3	0.84(0.23)	**0.33**(0.01)	0.38(0.02)
Test 4	0.74(0.11)	1.08(1.72)	**0.55**(0.47)
Mean	1.37(1.12)	0.78(1.41)	**0.45**(0.57)

RealSense D435i is employed to gather visual data, solely to record experimental scenarios in this work. For outdoor scenarios, a Bynav RTK GNSS [18] is used to acquire ground truth with an absolute positioning accuracy of 2 [cm], and the related transformation is manually calibrated based on the CAD model of the sensor suite and the A1 robot.

B. Dataset and Results

We collected the data on the main campus of HUST, as illustrated in Fig. 2. The dataset comprises four experiments conducted in different environments. In Test 1, the robot navigates on the cement road of the HUST square, with no significant slippage occurring between the robot feet and the ground. In Test 2, the robot walks on a pebble path in the forest park, where there are sparse trees and a few pedestrians. In Test 3, the robot traverses a soft lawn, and due to the soil softness, the robot feet noticeably shift when making contact with the ground. In Test 4, the robot walks on the playground, which offers ample space with many people running. The ground truth is obtained by the Bynav RTK GNSS.

The qualitative results of our algorithm in Test 2 are presented in Fig. 5. As shown in the left of Fig. 5, our algorithm successfully reconstructs a precise and dense 3D map of the forest park in real-time. Fig. 5 also demonstrates the highly accurate odometry computed by our algorithm. Please see our complementary video [8] for more details.

To evaluate the quantitative performance of our method, the mean relative position error (RPE) is computed over distances of 10 meters in Test $1-4$. Our algorithm is compared with two baselines. The first baseline is Fast-LIO2 [4] with the official recommended parameters, and the second baseline is a Kalman filter based legged odometry method [19], denoted as LEO. We excluded visual-based methods in the comparison, because we found that they demonstrated

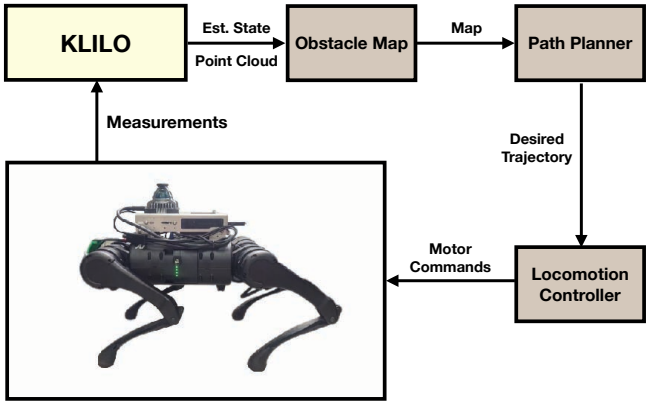

Fig. 6. The navigation framework. Our KLILO algorithm leverage the sensor measurements on the robot to calculate the estimated state and the registered point cloud in the global map. Then the occupancy map is updated and sent to the path planner. The planned trajectory is sent to an MPC based locomotion controller [20].

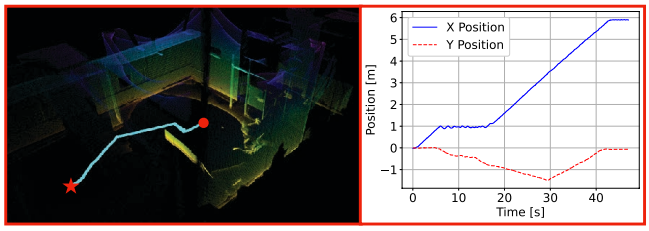

Fig. 7. Qualitative results of the navigation test. (left) The dense map constructed in real-time. The computed trajectory is colored in blue. The start and end positions are highlighted as the red circle and star, respectively. (right) The travel positions during navigation.

poor performance for the evening tests in the dataset.

The comparative results are presented in Table I. Our algorithm achieves the highest accuracy in all tests except Test 3. In Test 3, because the terrain is soft and deformable, the legged measurement model is inaccurate. Consequently, fusing legged kinematics information cannot improve the positioning accuracy. However, our method is still much more accurate than LEO. Also, the average RPE of our algorithm is the least compared to the baselines.

To summarize, we can conclude that our method is beneficial from the tight integration of LiDAR, inertial, and legged kinematics, and it outperforms LIO and legged odometry. Notably, the most significant improvement over Fast-LIO2 is observed in Test 4. In this scenario, the playground is spacious, where there are many dynamic objects (see Fig. 2). This poses potential challenges for LiDAR based odometry.

C. Navigation Application

The local terrain map around the robot plays an important role in perceptive locomotion and path planning. To further evaluate our algorithm in real-world applications, we integrated KLILO with a cascade navigation framework on the real robot. As shown in Fig. 6, the navigation framework includes a 2D path planner and a locomotion controller.

For our evaluation, we conduct a test where we initially place the robot in front of a wall and set a goal position

$(6, 0)$ beyond the wall. After initialization, we place a shade cloth on the robot during navigation, as shown in Fig. 1. It is worth noting that the dense map has been constructed in the initialization stage and can be used for path planning and locomotion control.

The qualitative demonstration of this test can be visualized in Fig. 7. This test is challenging for Fast-LIO2 or other exteroceptive algorithms, because most of the LiDAR points become ineffective after the shade cloth is put onto the robot. In contrast, our multi-sensor fusion algorithm remains effective and exhibits robust performance. The robot positions in x and y axises in the global frame can be visualized in the right figure of Fig. 7. It can be seen that the robot can successfully navigate to the final goal position while avoiding collisions.

Summarize the results from Section IV-B and IV-C, and we can conclude that KLILO is a practically viable solution for real-world applications.

V. Conclusion

This paper presents KLILO, a Kalman filter based state estimation algorithm for legged robots. In particular, the proposed method tightly fuses the measurements of the IMU, LiDAR, and legged kinematics within a manifold-based iterated error-state extended Kalman filter framework. Through extensive experiments, our odometry system outperforms individual sensor based solutions, showcasing the best overall performance in challenging environments. Moreover, our proposed algorithm is validated by a challenging navigation task on a real robot.

This paper has potential for extension in several directions. Firstly, it would be beneficial to further incorporate slippage and deformations into the legged measurement model. Moreover, the framework could be further developed to include visual sensors, particularly in environments where both LiDAR and legged kinematics may encounter limitations or failures.

Acknowledgement

This work is supported under the National Natural Science Foundation of China under Grant 62173155 and the Program for HUST Academic Frontier Youth Team.

References

[1] 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," *Robotics*, vol. 17, pp. 17–24, 2013.

[2] 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.

[3] 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.

[4] 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.

[5] 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.

[6] T. Qin, P. Li, and S. Shen, "Vins-mono: A robust and versatile monocular visual-inertial state estimator," *IEEE Transactions on Robotics*, vol. 34, no. 4, pp. 1004–1020, 2018.

[7] D. He, W. Xu, and F. Zhang, "Symbolic representation and toolkit development of iterated error-state extended kalman filters on manifolds," *IEEE Transactions on Industrial Electronics*, 2023.

[8] Klilo: Kalman filter based lidar-inertial-leg odometry for legged robots. [Online]. Available: https://www.xushaohang.top/home/klilo

[9] 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.

[10] 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.

[11] 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.

[12] 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.

[13] 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.

[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] B. M. Bell and F. W. Cathey, "The iterated kalman filter update as a gauss-newton method," *IEEE Transactions on Automatic Control*, vol. 38, no. 2, pp. 294–297, 1993.

[16] Unitree A1, https://www.unitree.com/products/a1/.

[17] F. Zhu, Y. Ren, and F. Zhang, "Robust real-time lidar-inertial initialization," in *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2022, pp. 3948–3955.

[18] Bynav, https://www.bynav.com/.

[19] T. Flayols, A. Del Prete, P. Wensing, A. Mifsud, M. Benallegue, and O. Stasse, "Experimental evaluation of simple estimators for humanoid robots," in *2017 IEEE-RAS 17th International Conference on Humanoid Robotics (Humanoids)*. IEEE, 2017, pp. 889–895.

[20] S. Xu, L. Zhu, H.-T. Zhang, and C. P. Ho, "Robust convex model predictive control for quadruped locomotion under uncertainties," *IEEE Transactions on Robotics*, 2023.