Invariant Smoother for Legged Robot State Estimation With Dynamic Contact Event Information

Ziwon Yoon ⓘ, Joon-Ha Kim ⓘ, *Graduate Student Member, IEEE*, and Hae-Won Park ⓘ, *Member, IEEE*

Abstract—This article proposes an invariant smoother for legged robot state estimation with the measurement of an inertial measurement unit and leg kinematics while assuming static foot contact. Because the proposed smoother is formulated with the residual functions with group-affine property, their Jacobians become independent from current state estimates. These state-independent Jacobians lead to better convergence properties in optimizing the cost in the smoother, especially under dynamic contact events. The proposed *slip rejection* method increases the uncertainty of static contact assumption when the robot has dynamic contact events. The estimated foot velocity, which is utilized to detect the dynamic contact events, is re-evaluated within the preserving time window. We also propose the *contact loop* method, a new measurement model asserting that foot position remains constant over multiple timesteps during stable contact. The proposed estimator is tested through online experiments, including indoor and 160 m-long outdoor experiments, and compared against state-of-the-art algorithms.

Index Terms—Dynamic contact event, legged robots, localization, sensor fusion.

I. INTRODUCTION

LEGGED robots are useful in complex environments since they can overcome uneven terrain with their legs. In order to control legged robots to perform dynamic locomotion, robust estimation of their state is essential, which typically includes the robot's orientation, position, and velocity [1], [2]. Therefore, algorithms have been developed to figure out the states of robots by exploiting various sensor measurements.

Considering not only the measurements themselves but also their noise properties, the state-estimation problem can be defined as the maximum a posteriori (MAP) problem, which maximizes the posterior probability distribution of the state to be estimated under the measured sensor values [4], [5], [6], [7]. By utilizing the entire history of states and measurements, batch

smoothers can solve the MAP problem and produce precise estimations, as they can relinearize potentially nonlinear probability distributions [8], [9], [10], [11], [12]. However, the growing size of the problem makes this approach unachievable in many real-world applications. As an alternative, many researchers use fixed-lag smoothers and filtering as a practical approximation, which can still provide accurate state estimation. Both algorithms preserve only a few latest states and measurements by marginalizing out the oldest, where the fixed-lag smoother keeps its states as many as its window size [13], [14], [15], [16], [17], [18], and [19], whereas the filter holds only the two latest states [20], [21], [22], [23], and [24]. Especially for filters, linearization of the nonlinear probability distribution is generally conducted only once, whereas smoothers relinearize nonlinear distributions multiple times. Furthermore, preserving more history can have benefits when measurements related to states at different times exist [14], [25], [26] or when the measurement probability distribution is dependent on the state itself [18].

Besides how long the history is preserved, selecting sensors for measurement is another design choice in the state-estimation problem. Sensors used for state estimation can be classified by the source from which a sensor obtains the information. One category is proprioceptive sensors, which collect information from the robot itself, including inertial measurement units (IMU), joint encoders, and torque/force measurements. Another is exteroceptive sensors, which receive information from the robot's surroundings, including the camera, LiDAR, and GPS signals. Previous works, including visual odometry algorithms [8], [9], [10], [13], [14], [15] and legged robot state estimators [11], [16], [17], [19], [20], [22], [23], [24], [27], have fused both types of sensors to provide a precise estimation of the state of the robot. Although fusing information from both proprioceptive and exteroceptive sensors gives the estimator richer information than using one kind, exteroceptive sensors are susceptible to suffering due to many unfavorable circumstances. For example, camera sensor measurement can be corrupted when the surface of the objects is reflective or transparent, and furthermore, feature-based odometry algorithms are prone to failure in featureless environments. Consequently, an estimator producing sufficiently accurate results without using exteroceptive sensors is necessary [12], [18], [21].

In the case of legged robots, IMU and joint encoders are widely used proprioceptive sensors, where joint encoder measurements are employed via an assumption that the foot in contact does not move [1], [16], [20]. However, the fixed contact assumption is occasionally violated in the real world due to *dynamic contact events,* such as slippage of the foot and stepping on unstable ground [17], [28], [29], where one example scenario is introduced in Fig. 1. For this reason, Bloesch et al. [28]

Fig. 1. Example scenario when static contact assumption of the foot is violated. A strong pulling force is applied to the quadruped robot Hound [3] while walking on a terrain made with moving wood piles and pebbles.

ignored the kinematic measurements that violate the fixed contact assumption by thresholding the Mahalanobis distance of the innovation. In our previous work [18], we unfolded the idea to a fixed-lag smoother to cope with the nonlinear dynamics of the system. Although the idea accounts for rejecting various potential violations of the contact foot assumption, we named it slip rejection (SR) to ease the intuitive understanding of the algorithm.

Meanwhile, Barrau and Bonnabel, [30], [31] introduced the invariant extended Kalman filter (InEKF), which shows the faster convergence of the observable states than the conventional extended Kalman filter [1] when an initial error exists by using *group-affine* property. Hartley et al. [32] discovered that the proprioceptive legged robot state estimation system is approximately group-affine and, hence, proposed the InEKF. Furthermore, invariant smoother (IS), a so-called name of the estimation algorithm that utilizes the properties of the group-affine property system in a smoother framework, was proposed by Chauchat et al. [33].

In summary, although there were studies about legged robot state estimation and group-affine property, an algorithm that thoroughly considers the profit of the group-affine property and smoothing framework specialized to legged robots has not been proposed. Therefore, in this work, we propose an invariant smoother for legged robot state estimation, which utilizes the state invariance of the group-affine property and the history-preserving smoother frameworks together. Moreover, we propose a new type of temporal displacement cost model for the legged robot system with the name of *contact loop* (CL), which restricts the estimated foot position to be fixed when the foot is in stable contact for longer than two timesteps. Thanks to the smoother framework, the proposed algorithm can not only consider temporal displacement costs but also increase the merit of the SR method since the state history used for detecting slippery contact can be re-evaluated, which is unachievable in filters. The resulting overall structure of our proposed algorithm is described in Fig. 2.

The main contributions of this article are as follows.

1) We propose an IS for legged robot state estimation, which is robust to dynamic contact events and external disturbances without using exteroceptive sensors.
2) The SR method, which has been developed by Kim et al. [18], is modified to apply a large uncertainty when a robot slips, where the previous method rejected the measurements. Furthermore, the contact event state is re-evaluated within the history window, which was fixed as the initial guess in the previous method.
3) We introduce a new measurement model for legged robot state estimation and name it the CL method, which reflects our belief that the foot position should be constant over multiple timesteps if the foot is in stable contact during that time. Since this method assumes that the foot position states of distant timesteps within the history window are the same, it cannot be utilized in the estimators that marginalize recent states like InEKF [32].
4) Finally, a quadruped robot *KAIST HOUND* [3] was controlled in the real world using the real-time feedback of the developed IS in various environments. With this experimental data, the benefit of the proposed algorithm was validated by comparing it against the InEKF [32] and *noninvariant Smoother* (*NIS*) [18].

The rest of this article is organized as follows. Section II introduces the related works of this article. Section III introduces the preliminaries for the article. Section IV proposes the IS for the legged robot system. Section V introduces the SR and CL methods. Section VI, then, describes the various alternative state definitions used in this article for comparison. Section VII shows the real-world experimental results with comparison against the existing algorithms (InEKF [32], NIS [18]), where the estimated states are used to control the quadruped robot Hound [3]. Finally, Section VIII concludes this article and introduces future work.

II. RELATED WORKS

In this section, we introduce some notable works that have been the base of our work. We first present the literature review of the previous works related to the group-affine property and then review how the legged robot state estimation field has been developed.

A. Group-Affine Property and Invariant Smoothers

The group-affine property, which was proposed by Barrau et al. [30], is a property that applies to a particular form of state variables and its related equations. The property has its name since a system that satisfies this property offers an *affine* function (a linear system) for the lie algebra error (usually called as *log invariant error*) of the state variables, which live in *Lie group*. With this useful property, one can model a linear propagation and observation equation for the system of this log invariant error, whereas its original group state variable has a nonlinear system. Leveraging this linearity, Barrau et al. [30] proposed an InEKF that shows impressive convergence performance regardless of the linearization point of the filter. Since the methodology applies only to the propagation and observation models that satisfy the group-affine property, many researchers have tried to prove that the system under consideration is group-affine to exploit the advantage.

Afterward, Chauchat et al. [33] further developed the InEKF to an invariant smoothing framework, which applies the group-affine property in a smoother framework. The proposed framework was applied in an exemplary system of 2-D position estimation of mobile cars equipped with gyroscopes, velocity odometry, and GNSS measurements. On the other hand, Walsh et al. [34] tested an IS for attitude and heading reference system with gyroscope bias in simulation. Also, Huai et al. [35] applied
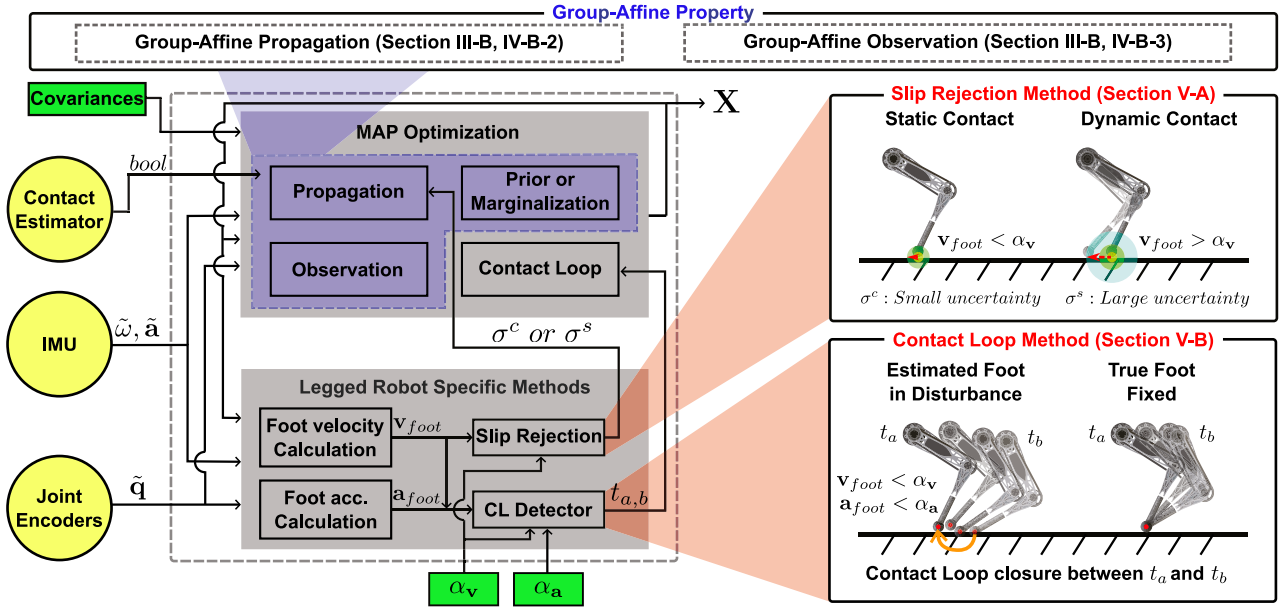
Fig. 2. Overall structure of the proposed algorithm. The estimator solves the MAP problem with the aid of the *group-affine* property in a nonlinear optimization manner while utilizing IMU, joint encoders, and estimated contact from sensors and contact estimator. Thanks to the group-affine property, the estimator can mitigate the suffering of different linearization points in nonlinear optimization and increase the estimation accuracy. Meanwhile, the algorithm qualifies the given sensor measurements by leveraging the special nature of legged robots. The *SR* method checks the violation of the static foot contact assumption based on the foot velocity and acceleration calculated by IMU, joint encoders, and the previously estimated state. The *CL* method restricts the estimated foot position to be fixed over a long time by formulating a temporal displacement cost between the chosen timesteps.

the group-affine property to the visual-inertial SLAM application of micro aerial vehicles and examined the application with simulation and public datasets.

B. Legged Robot State Estimation

Control of a legged robot requires accurately estimating its key state variables, such as orientation and velocity. While most legged robots are equipped with proprioceptive sensors, such as IMU and joint encoders, these sensors cannot directly measure orientation and velocity.

Nevertheless, Bloesch et al.[1] proposed an extended Kalman filter-based state-estimation algorithm by assuming that a robots' foot in contact does not move and uses the joint encoder measurement as a kinematic odometry provider. Upon this work, the idea of kinematic odometry has been leveraged in various estimator frameworks, including unscented Kalman filter [28], fixed-lag smoother [18], [23], and InEKF [32]. Since the kinematic odometry originated from the static contact foot assumption, the contact state of the foot should be detected from a contact sensor or estimated from the other kinds of sensors. Unfortunately, many existing legged robots [3], [36] do not have an explicit contact-detecting sensor on their feet. Therefore, legged robot state estimators rely on model-based contact detection algorithms [37], [38], or learning-based approach [39].

Although these contact detectors are often reliable, estimation results easily diverge when this static foot contact assumption is violated. To cope with this drawback, the authors in [18] and [28], developed the SR method, which denies the kinematic odometry when the difference between the kinematic odometry and the IMU odometry exceeds a certain threshold. Since this method made robots less susceptible to dynamic contact events, the method has been applied to various estimator frameworks,

such as fixed-lag smoother [18] and InEKF [24]. Unlike the previously introduced outlier rejection methods, there was also a constraint Kalman filter-based algorithm that takes into account the linear complementarity condition for the contact implicit dynamics relationship to estimate the state of the rigid body dynamics, including the contact state of the robot [40]. However, the algorithm assumes that the terrain properties, like the slope of the ground and friction coefficients, are previously known, which may be challenging to identify in the real world.

Other than the estimation frameworks that can deal with the quality of the contact estimation, there were also estimator frameworks that deal with the uncertainty of the kinematic odometry itself, such as kinematic chain uncertainty modeling of legs [41] and learning-based odometry provider [42].

On the other hand, researchers have recently been using exteroceptive sensors for legged robots to provide precise positional estimation and perform SLAM. The authors in [27] and [19] used LiDAR or a depth camera with proprioceptive sensors and conducted long-range estimation tests.

III. PRELIMINARIES

This section explains mathematical preliminaries that are essential for understanding this article: matrix Lie groups, the group-affine property, and the smoothing framework.

A. Matrix Lie Groups

Most of the robot state estimation works over the orientation of the robot body. However, the 3-D rotation matrix, which represents the orientation of the robot, is difficult to handle since it does not live in vector spaces but on a manifold.

Specifically, the rotation matrix forms a matrix Lie group called *special orthogonal group* SO(3) [7], [43], [44], [45], [46]. In addition, dealing with the position and the velocity of the robot often requires knowledge of a matrix Lie group called *special Euclidean group* $\mathrm{SE}_k(3)$. Therefore, this section covers useful properties of matrix Lie group SO(3) and $\mathrm{SE}_k(3)$ to enhance the understanding of the characteristics of manifold variables.

1) Special Orthogonal Group SO(3): The 3-D rotation matrix $\mathbf{R}$ belongs to a 3-D special orthogonal group [SO(3) group], which is a three-by-three orthogonal real matrix that satisfies the following relationships:

$$\mathrm{SO}(3) \ : \ \{\mathbf{R} \in \mathbb{R}^{3\times 3} : \ \mathbf{R}^T\mathbf{R} = \mathbb{I}_3, \ \det(\mathbf{R}) = 1\} \quad (1)$$

where $\mathbb{I}_{sx} \in \mathbb{R}^{sx \times sx}$ is the sx dimensional identity matrix.

SO(3) group is known to have a mapping between the group elements and its *Lie algebra* $\mathfrak{so}(3) \in \mathbb{R}^{3\times 3}$ and a tangential vector space $\phi \in \mathbb{R}^3$. For SO(3) group, the mapping from the algebra and the group is the matrix exponential $\exp(\cdot)$, and between the vector space and the algebra is the *hat* operation $(\cdot)^\wedge$, which is a skew-symmetric matrix for an input vector

$$\phi^\wedge = \begin{bmatrix} \phi_x \\ \phi_y \\ \phi_z \end{bmatrix}^\wedge = \begin{bmatrix} 0 & -\phi_z & \phi_y \\ \phi_z & 0 & -\phi_x \\ -\phi_y & \phi_x & 0 \end{bmatrix} \in \mathbb{R}^{3\times 3}. \quad (2)$$

The inverse mapping is the matrix logarithm $\log(\cdot)$ and the *vee* operation $(\cdot)^\vee$ and the resulting overall relationships are summarized as follows:

$$\mathrm{Exp}(\phi) := \ \exp(\phi^\wedge) \in \mathrm{SO}(3) \leftarrow \phi^\wedge \in \mathfrak{so}(3) \leftarrow \phi \in \mathbb{R}^3 \quad (3)$$

$$\mathrm{Log}(\mathbf{R}) := \ \log(\mathbf{R})^\vee \in \mathbb{R}^3 \leftarrow \log(\mathbf{R}) \in \mathfrak{so}(3) \leftarrow \mathbf{R} \in \mathrm{SO}(3). \quad (4)$$

For convenience, we added the capitalized exponential and logarithmic mapping, $\mathrm{Exp}(\cdot)$ and $\mathrm{Log}(\cdot)$, which directly maps the vector space variable to/from the Lie group variable, respectively. Based on the definition, we introduce useful properties of these operations from [6], which would abbreviate the rest of our discussion.

$$\mathrm{Exp}(\delta\phi)\mathrm{Exp}(\phi_1) \approx \mathrm{Exp}(\mathbf{J}_l^{-1}(\phi_1)\delta\phi + \phi_1) \quad (5)$$

$$\mathrm{Log}\,(\mathrm{Exp}(\delta\phi)\mathrm{Exp}(\phi_1)) \approx \mathbf{J}_l^{-1}(\phi_1)\delta\phi + \phi_1 \quad (6)$$

$$\mathrm{Exp}(\delta\phi) \approx \mathbb{I}_3 + (\delta\phi)^\wedge \quad (7)$$

$$\phi_1^\wedge \phi_2 = -\phi_2^\wedge \phi_1. \quad (8)$$

Here, $\delta\phi$, ϕ_1, and ϕ_2 are arbitrary vectors in $\mathbb{R}^3$ and $\mathbf{J}_l(\cdot)$ is the left Jacobian of the SO(3) manifold [6]. Equations (5) and (6) are called *Baker–Campbell–Hausdorff (BCH) formula*, which approximate Lie group operation to vector space operation when $\delta\phi$ is small. Equation (7) is the first-order approximation of the Taylor expansion of the exponential map. Finally, (8) is the property of skew-symmetric matrices. The detailed proofs are in [6] and are not repeated here.

2) Special Euclidean Group $SE_k(3)$: The element of $\mathrm{SE}_k(3)$ is a square matrix of size $(3 + k)$ by $(3 + k)$, constituted by a rotation matrix $\mathbf{R} \in \mathrm{SO}(3)$ and k number of vectors $^1\mathbf{p}, \dots, ^k\mathbf{p} \in$ $\mathbb{R}^3$, as below:

$$\mathbf{X} := \begin{bmatrix} \mathbf{R} & ^1\mathbf{p} & \cdots & ^k\mathbf{p} \\ \mathbf{0}_{1,3} & 1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{0}_{1,3} & 0 & \cdots & 1 \end{bmatrix} \in \mathrm{SE}_k(3) \quad (9)$$

where $\mathbf{0}_{sx,sy} \in \mathbb{R}^{sx \times sy}$ is the sx by sy zero matrix. Akin to SO(3), variable $\mathbf{X} \in \mathrm{SE}_k(3)$ is often mapped to a Euclidean vector space variable $\xi := [\phi^T, ^1\xi^T, \dots, ^k\xi^T]^T \in \mathbb{R}^{3+3\,k}$, using the exponential map and logarithmic map, which directly maps to/from vector space variable, for $^i\xi \in \mathbb{R}^3$

$$\xi \in \mathbb{R}^{3+3\,k} \ \rightarrow \ \mathrm{Exp}(\xi) \in \mathrm{SE}_k(3) \quad (10)$$

$$\mathbf{X} \in \mathrm{SE}_k(3) \ \rightarrow \ \mathrm{Log}(\mathbf{X}) \in \mathbb{R}^{3+3\,k} \quad (11)$$

where,

$$\mathrm{Exp}(\xi) = \exp(\xi^\wedge)$$

$$= \begin{bmatrix} \exp(\phi^\wedge) & \mathbf{J}_l(\phi)(^1\xi) & \cdots & \mathbf{J}_l(\phi)(^k\xi) \\ \mathbf{0}_{1,3} & 1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{0}_{1,3} & 0 & \cdots & 1 \end{bmatrix}. \quad (12)$$

Similar to the SO(3) manifold, the logarithmic map is the inversion of the exponential map. Note that when $k = 0$, the mapping functions are identical with the SO(3) case. In addition, the hat operation on ξ is

$$\xi^\wedge = \begin{bmatrix} \phi^\wedge & ^1\xi & \cdots & ^k\xi \\ \mathbf{0}_{1,3} & 0 & \cdots & 0 \\ \mathbf{0}_{1,3} & \vdots & \ddots & \vdots \\ \mathbf{0}_{1,3} & 0 & \cdots & 0 \end{bmatrix} \in \mathbb{R}^{(3+k)\times(3+k)}. \quad (13)$$

In addition, we introduce the matrix adjoint $\mathrm{Ad}_\mathbf{X} \in \mathbb{R}^{(3+3\,k)\times(3+3\,k)}$ of $\mathrm{SE}_k(3)$, which carries a considerable part in our discussion.

$$\mathrm{Ad}_\mathbf{X} = \begin{bmatrix} \mathbf{R} & \mathbf{0}_{3,3} & \cdots & \mathbf{0}_{3,3} \\ ^1\mathbf{p}^\wedge\mathbf{R} & \mathbf{R} & \cdots & \mathbf{0}_{3,3} \\ \vdots & \vdots & \ddots & \vdots \\ ^k\mathbf{p}^\wedge\mathbf{R} & \mathbf{0}_{3,3} & \cdots & \mathbf{R} \end{bmatrix}. \quad (14)$$

Parallel to SO(3), useful approximations for $\mathrm{SE}_k(3)$ are given from [6]

$$\mathrm{Exp}(\delta\xi)\mathrm{Exp}(\xi_1) \approx \mathrm{Exp}(\mathbf{J}_l^{-1}(\xi_1)\delta\phi + \xi_1) \quad (15)$$

$$\mathrm{Log}\,(\mathrm{Exp}(\delta\xi)\mathrm{Exp}(\xi_1)) \approx \mathbf{J}_l^{-1}(\xi_1)\delta\phi + \xi_1 \quad (16)$$

$$\mathrm{Exp}(\delta\xi) \approx \mathbb{I}_{(3+k)} + (\delta\xi)^\wedge \quad (17)$$

where, (15) and (16) are the BCH formula of the $\mathrm{SE}_k(3)$ group. Finally, a similar equation with (8) for hat operation in $\mathrm{SE}_k(3)$ is presented

$$\xi_1^\wedge \xi_2 = \xi_2^\odot \xi_1 \quad (18)$$

where ξ_1 and ξ_2 are arbitrary vectors in $\mathbb{R}^{3+k}$ and $\xi^\odot \in \mathbb{R}^{(3+k)\times(3+3k)}$ is defined as

$$\xi^\odot := \begin{bmatrix} -\phi^\wedge & {}^1\xi \cdot \mathbb{I}_3 & \cdots & {}^k\xi \cdot \mathbb{I}_3 \\ \mathbf{0}_{1,3} & \mathbf{0}_{1,3} & \cdots & \mathbf{0}_{1,3} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{0}_{1,3} & \mathbf{0}_{1,3} & \cdots & \mathbf{0}_{1,3} \end{bmatrix}. \quad (19)$$

B. Group-Affine Property

In this section, we will briefly introduce the group-affine property, which contains useful properties that make the observer state-independent [30], [33]. The group-affine property comes from a particular type of propagation and observation model defined on the matrix Lie group.

First, in the case of the propagation model, for arbitrary variables $\mathbf{X}_t$ and $\bar{\mathbf{X}}_t$ living on the same Lie group $\mathbb{G}$, the left-invariant error η_t^l at time t and the right-invariant error η_t^r at time t can be defined between the two variables as follows:

$$\eta_t^l := \mathbf{X}_t^{-1}\bar{\mathbf{X}}_t \quad \text{(left-invariant)}$$
$$\eta_t^r := \bar{\mathbf{X}}_t\mathbf{X}_t^{-1} \quad \text{(right-invariant)} \quad (20)$$

where the terminology of the left and right invariant errors originate from the invariance of the left multiplication $(\mathbf{X}_t, \bar{\mathbf{X}}_t) \to (\mathbf{Y}\mathbf{X}_t, \mathbf{Y}\bar{\mathbf{X}}_t)$ and the right multiplication $(\mathbf{X}_t, \bar{\mathbf{X}}_t) \to (\mathbf{X}_t\mathbf{Y}, \bar{\mathbf{X}}_t\mathbf{Y})$, respectively, about an arbitrary variable $\mathbf{Y}$ on the group $\mathbb{G}$. Furthermore, the left log-invariant error ξ_t^l and the right log-invariant error ξ_t^r at time t are defined by taking the logarithmic mapping of the left and right invariant errors, respectively.

$$\xi_t^l := \text{Log}(\eta_t^l) = \text{Log}(\mathbf{X}_t^{-1}\bar{\mathbf{X}}_t) \quad \text{(left-invariant)}$$
$$\xi_t^r := \text{Log}(\eta_t^r) = \text{Log}(\bar{\mathbf{X}}_t\mathbf{X}_t^{-1}) \quad \text{(right-invariant)}. \quad (21)$$

Now, for a system whose propagation function $f(\cdot)$ satisfies the following condition, the system is called to be *group-affine* [30].

$$\frac{\mathrm{d}}{\mathrm{d}t}\mathbf{X}_t = f(\mathbf{X}_t), \; f(\mathbf{X}\mathbf{Y})$$
$$= \mathbf{X}f(\mathbf{Y}) + f(\mathbf{X})\mathbf{Y} - \mathbf{X}f(\mathbb{I}_{\dim(\mathbf{Y})})\mathbf{Y}.$$
$$for \quad \forall t \geq 0 \quad \forall \mathbf{X}, \mathbf{Y} \in \mathbb{G}. \quad (22)$$

Assuming that two trajectories $\mathbf{X}_{0:t}$ and $\bar{\mathbf{X}}_{0:t}$ are proceeded by the same group-affine propagation $f(\cdot)$, it is proved in [30] that there is a function $g(\cdot)$ that can describe the propagation of the invariant error defined between the trajectories

$$\frac{\mathrm{d}}{\mathrm{d}t}\eta_t = g(\eta_t)$$
$$g(\eta_t) = f(\eta_t) - f(\mathbb{I}_{\dim(\mathbf{Y})})\eta_t \quad \text{(left-invariant)}$$
$$f(\eta_t) - \eta_t f(\mathbb{I}_{\dim(\mathbf{Y})}) \quad \text{(right-invariant)}. \quad (23)$$

Moreover, in [30], it is proven that there is a constant matrix $\mathbf{G}$, which describes the linear propagation of the log-invariant error in vector spaces as follows:

$$g(\eta_t) = (\mathbf{G}\xi_t)^\wedge + O(\|\xi_t\|^2), \quad \frac{\mathrm{d}}{\mathrm{d}t}\xi_t = \mathbf{G}\xi_t. \quad (24)$$

As the above equations hold for both left- and right-invariant errors, we drop out the superscript and only cover the right-invariant case without loss of generality. Since the matrix $\mathbf{G}$
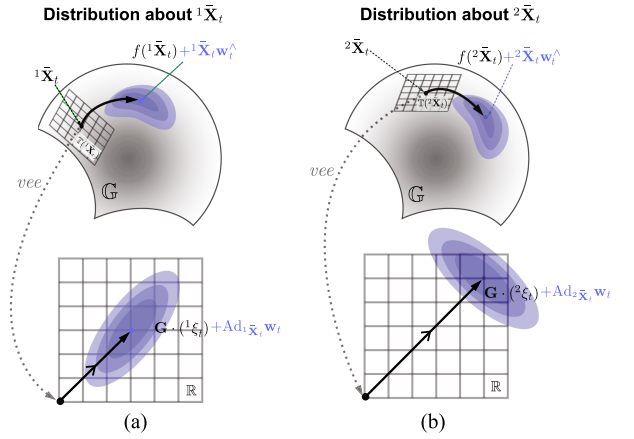

Fig. 3. Graphical illustration of the propagation model, which satisfies the group-affinity. The tangential space around the state $\mathbf{X}$ is denoted as $\mathbb{T}(\mathbf{X})$. Employing the identical noise vector $\mathbf{w}_t$, the uncertainty distribution (visually depicted as a color-gradating ellipse) preserves its shape across distinct operating points ${}^1\bar{\mathbf{X}}_t$ [subfigure (a)] and ${}^2\bar{\mathbf{X}}_t$ [subfigure (b)] within the perturbed vector space. Notably, their sole distinction is the coordinate transformation from the adjoint matrix.

is constant, the evolution of the error is autonomous (state-independent), regardless of the two trajectories $\mathbf{X}_{0:t}$ and $\bar{\mathbf{X}}_{0:t}$. This is called *log-linear property of the error* [30]. In addition, we bring the noise corrupted version of (22) and (24) from [30]

$$\frac{\mathrm{d}}{\mathrm{d}t}\mathbf{X}_t = f(\mathbf{X}_t) + \mathbf{X}_t(\mathbf{w}_t^{\text{Prop}})^\wedge$$
$$\frac{\mathrm{d}}{\mathrm{d}t}\xi_t = \mathbf{G}\xi_t - \text{Ad}_{\bar{\mathbf{X}}_t}\mathbf{w}_t^{\text{Prop}} \quad (25)$$

where $\text{Ad}_{\bar{\mathbf{X}}_t}$ is the adjoint matrix of $\bar{\mathbf{X}}_t$ and $\mathbf{w}_t^{\text{Prop}}$ is the uncertainty of the propagation model. The graphical illustration of the right-invariant propagation model is given in Fig. 3. Note that the propagation model, which satisfies the group-affinity, makes the original nonlinear distribution to the Gaussian distribution at the perturbed vector space about ξ_t. Furthermore, the shape of the distribution is state-independent even when the current operating point changes from ${}^1\bar{\mathbf{X}}_t$ to ${}^2\bar{\mathbf{X}}_t$ and only the coordinate transformation through the adjoint matrix appears differently. This state-independent distribution shape resulting from the group-affine property can be beneficial for state-estimation problems compared with the state-dependent distributions, which can suffer problems when the current estimated state becomes far from the true state.

Although the prior discussions focus on the continuous domain, it is necessary to derive discrete equations for formulating the optimization method since it is fundamentally defined in discrete terms. As a result, we introduce the following propagation equation in the discrete domain:

$$\mathbf{X}_{i+1} = \text{Exp}(-\xi_{i+1})\bar{\mathbf{X}}_{i+1} \approx \text{Exp}(-g_d(\xi_i, \Delta t))f_d(\bar{\mathbf{X}}_i, \Delta t) \quad (26)$$

where i is the discretized timestep. In this equation, the error dynamics tainted by noise and the deterministic dynamics of the state propagation are separated and then discretized individually using $g_d(\cdot)$ and $f_d(\cdot)$. This approach approximates the exact log-linearity directly derived from the discrete domain, which has been exploited in [31] and [33]. However, it provides a useful

way to get discrete-domain equations rooted in the continuous-domain group-affine property when it is challenging to obtain the discrete-domain group-affine propagation equation.

Second, in the case of the observation model, compared with the propagation model, it is straightforward that when the observation $\mathbf{y}_i$ at timestep i takes the following form, the model satisfies group-affinity:

$$\mathbf{y}_i = \mathbf{X}_i \mathbf{s} + \mathbf{w}_i^{\text{Obs}} \qquad \text{(left-invariant)} \qquad (27)$$

$$\mathbf{y}_i = \mathbf{X}_i^{-1} \mathbf{s} + \mathbf{w}_i^{\text{Obs}} \qquad \text{(right-invariant)} \qquad (28)$$

where $\mathbf{s}$ is a constant vector and $\mathbf{w}_i^{\text{Obs}}$ is Gaussian noise vector associated with the observation model [30]. Left-multiplying $\bar{\mathbf{X}}_i$ on both sides of the right-invariant observation model (28), the model becomes

$$\bar{\mathbf{X}}_i \mathbf{y}_i = \text{Exp}(\xi_i)\mathbf{s} + \bar{\mathbf{X}}_i \mathbf{w}_i^{\text{Obs}}. \qquad (29)$$

Assuming that ξ_i is small, we can linearize the equation with the first-order approximation of the exponential map (17)

$$\bar{\mathbf{X}}_i \mathbf{y}_i = \mathbf{s} + \xi_i^{\wedge}\mathbf{s} + \bar{\mathbf{X}}_i \mathbf{w}_i^{\text{Obs}}. \qquad (30)$$

This linearized model can be reorganized to explicitly show the Jacobian of ξ_i by using (18).

$$\bar{\mathbf{X}}_i \mathbf{y}_i = \mathbf{s} + \mathbf{s}^{\odot}\xi_i + \bar{\mathbf{X}}_i \mathbf{w}_i^{\text{Obs}}. \qquad (31)$$

Although the above derivation takes place for the right-invariant observation, the same process is available for the left-invariant observation by multiplying $\bar{\mathbf{X}}_i^{-1}$ on the model (27). Similarly to the propagation model (25), the observation model satisfying the group-affine property gives a constant Jacobian $\mathbf{s}^{\odot}$, as (31) shows.

C. Smoothing Framework

A state-estimation problem can be regarded as estimating the states that maximize the posterior probability between the states and measurements. This is also called a MAP problem and is defined as follows:

$$\chi_{0:n}^* = \text{argmax}_{\chi_{0:n}} \, p\left(\chi_{0:n} \mid \mathbf{Z}_{0:n}\right)$$

$$= \text{argmax}_{\chi_{0:n}} \, p(\chi_0)p(\mathbf{Z}_{0:n}|\chi_{0:n})$$

$$:= \text{argmax}_{\chi_{0:n}} \left(\prod_{j \in \boldsymbol{P}_n} p_j(\cdot) \right) \qquad (32)$$

where $p(\chi_0)$ is the prior probability distribution about the initial state χ_0, $\mathbf{Z}_{0:n}$ is the set of sensor measurements from the initial timestep 0 to the current timestep n, $\chi_{0:n}$ is the state from timestep 0 to n, $\boldsymbol{P}_n$ is a set of probability distributions related to the posterior probability from timestep 0 to n, and $p_j(\cdot)$ is the according jth distribution, which is assumed to be a *Gaussian measurement model* [47], [48], [49] as $p_j(\cdot) = \mathcal{N}(\mathbf{r}_j; \mathbf{0}_{m,1}, \boldsymbol{\Sigma}_j)$. Here, $\mathbf{r}_j \in \mathbb{R}^m$ is the residual function associated with the measurements and states, m is the size of the residual function, and $\boldsymbol{\Sigma}_j$ is the corresponding covariance matrix. The multivariate Gaussian distribution $\mathcal{N}(\mathbf{x}; \mu, \boldsymbol{\Sigma})$ above is defined as follows:

$$\mathcal{N}(\mathbf{x}; \mu, \boldsymbol{\Sigma}) = \frac{\exp(-\frac{1}{2}(\|\mathbf{x} - \mu\|_{\boldsymbol{\Sigma}}^2))}{\sqrt{(2\pi)^m det(\boldsymbol{\Sigma})}} \qquad (33)$$

where $\|\mathbf{x} - \mu\|_{\boldsymbol{\Sigma}}^2 := (\mathbf{x} - \mu)^T \boldsymbol{\Sigma}^{-1}(\mathbf{x} - \mu)$ is the squared Mahalanobis distance with $\mathbf{x}, \mu \in \mathbb{R}^m$ and $\boldsymbol{\Sigma} \in \mathbb{R}^{m \times m}$. Now, the

maximization problem can be further transformed into a negative log minimization problem as follows:

$$\chi_{0:n}^* = \underset{\chi_{0:n}}{\text{argmax}} \left(\prod_{j \in \boldsymbol{P}_n} p_j(\cdot) \right) = \underset{\chi_{0:n}}{\text{argmin}} - \log \left(\prod_{j \in \boldsymbol{P}_n} p_j(\cdot) \right)$$

$$= \underset{\chi_{0:n}}{\text{argmin}} \left(\sum_{j \in \boldsymbol{P}_n} \|\mathbf{r}_j\|_{\boldsymbol{\Sigma}_j}^2 \right). \qquad (34)$$

Note that the residual function $\mathbf{r}_j$ represents the difference between the observed value of the sensor model (with the measured sensor value as an input and the model's predicted values). In the case of the according covariance matrix $\boldsymbol{\Sigma}_j$, it represents the noise property of the residual function within the Gaussian distribution model.

Then, the Gauss–Newton algorithm is usually applied to this nonlinear least-square optimization problem, which optimizes the objective function about the perturbed states $\delta\chi_{0:n}$ with the iterative linearization at the current operating point $\bar{\chi}_{0:n}$ [6], [7], [50], [51].

$$\delta\chi_{0:n}^* = \text{argmin}_{\delta\chi_{0:n}} \left(\sum_{j \in \boldsymbol{P}_n} \|\bar{\mathbf{r}}_j - \bar{\mathbf{J}}_j \Delta\chi_j\|_{\boldsymbol{\Sigma}_j}^2 \right) := -\mathbf{H}^{-1}\mathbf{z} \qquad (35)$$

where $\bar{\mathbf{r}}_j$, $\bar{\mathbf{J}}_j$, and $\Delta\chi_j$ are the residual, Jacobian at the operating points, and concatenation of perturbed states related to the jth distribution, respectively. $\mathbf{H}$ and $\mathbf{z}$ are the large matrix and vector obtained by summing Hessian matrices $\bar{\mathbf{J}}_j^T \boldsymbol{\Sigma}_j^{-1} \bar{\mathbf{J}}_j$ and gradient vectors $\bar{\mathbf{J}}_j^T \boldsymbol{\Sigma}_j^{-1} \bar{\mathbf{r}}_j$ of all the consisting cost models [7]. $\delta\chi_i$ is defined to be the local perturbation of the current operating point $\bar{\chi}_i$ at each timestep i, and the optimal state χ_i^* is calculated based on the obtained Gauss–Newton perturbation $\delta\chi_{0:n}^*$ as follows [18], [50], [51]:

$$\chi_i^* \leftarrow \bar{\chi}_i + \delta\chi_i^*. \qquad (36)$$

This procedure is repeated until the optimization problem converge. However, the evaluated change is often excessive due to the erroneous linearization of the model. One of the possible methods to cope with this problem is Armijo's backtracking line search [52], which adjusts the step size by multiplying the tunable step size parameter α in front of the transition to prevent the failure of the Gauss–Newton method as follows:

$$\chi_i^* \leftarrow \bar{\chi}_i + \alpha\delta\chi_i^*. \qquad (37)$$

Although the backtracking approach prevents potential divergence, finding proper α generally requires iterative cost calculation and may entail additional computation time, which could impair real-time applicability. For offline experiments, decent α is found iteratively by gradually decreasing it from one, checking whether the state change with current α predicts the lower cost. If the predicted cost is higher than the current cost, current α decreases by $\gamma\alpha$ and then is retested. The decreasing rate γ and the number of iterations spent to find α are called the backtracking rate and backtracking number n_{back}, respectively [52].

The previously described Gauss–Newton algorithm cannot be directly applied when the incorporated optimization variables are defined on the manifold. However, by approximating the manifold operations locally to its tangent vector spaces of the current operation point, it is possible to approximate the MAP

problem to a nonlinear least-square problem in vector spaces as follows [10], [18]:

$$\delta\mathbf{X}_{0:n}^{*} = \text{argmin}_{\delta\mathbf{X}_{0:n}} \left(\sum_{j \in \boldsymbol{P}_n} \|\bar{\mathbf{r}}_j - \bar{\mathbf{J}}_j \Delta\mathbf{X}_j\|_{\bar{\boldsymbol{\Sigma}}_j}^2 \right)$$

$$\mathbf{X}_i^* \leftarrow \text{Exp}(\delta\mathbf{X}_i^*)\bar{\mathbf{X}}_i \qquad (38)$$

where $\delta\mathbf{X}_i$ is defined to be the local perturbation on the tangential space of the current operating state $\bar{\mathbf{X}}_i$ on the manifold at each timestep i, and $\Delta\mathbf{X}_j$ to be the concatenation of the perturbed local states related to the jth distribution. That is, we can locally parameterize manifold variables to vector space [2], [10], [27], [53], [54]. After solving the nonlinear optimization problem in vector space, the evaluated perturbation is retracted on the manifold to update the operating point.

In the following sections, these preliminaries are further expanded to an invariant fixed-lag smoothing framework while mediating between the accuracy and computation time by adjusting the window size.

IV. INVARIANT SMOOTHING FOR LEGGED ROBOT SYSTEM

In this section, we describe the invariant smoothing framework for the legged robot system, which leverages the group-affine property in a smoother framework. Since describing both left-invariant and right-invariant cases is redundant, our discussion will only cover the right-invariant case without loss of generality.

A. State and Measurement Definition

For legged robot state estimation, we focus on estimating the base orientation $\mathbf{R}$, velocity $\mathbf{v}$, and position $\mathbf{p}$ of the robot. Also, $^{i_{\text{leg}}}\mathbf{d}$, the position of the i_{leg}th foot in contact with the ground, and $\mathbf{b}^\omega$ and $\mathbf{b}^a$, which are IMU biases, are included in the state definition to consider the N-legged robot state estimation

$$\mathbf{X}_i := \begin{bmatrix} \mathbf{R}_i & \mathbf{v}_i & \mathbf{p}_i & ^1\mathbf{d}_i & \cdots & ^N\mathbf{d}_i \\ \mathbf{0}_{(N+2)\times 3} & & & \mathbb{I}_{(N+2)} & \end{bmatrix},$$

$$\mathbf{x}_i := \begin{bmatrix} \mathbf{b}_i^\omega \\ \mathbf{b}_i^a \end{bmatrix}. \qquad (39)$$

Inspired from [32], we define the states in the Lie group to take care of the group-affine property, whereas the IMU biases remain in vector spaces since their related equations are not group-affine. Detailed reasons for defining the state in this form will be discussed in the following sections. Now the perturbation of the system is defined as the concatenation of the right-invariant perturbation at $\bar{\mathbf{X}}$ and the vector space perturbation at $\bar{\mathbf{x}}$.

$$\xi_i = -\text{Log}(\bar{\mathbf{X}}\mathbf{X}^{-1}), \quad \zeta_i = \mathbf{x}_i - \bar{\mathbf{x}}_i,$$

$$\mathbf{e}_i = [\xi_i^T, \zeta_i^T]^T \in \mathbb{R}^{15+3N}. \qquad (40)$$

The minus sign in front of the invariant error is attached to match the sign of the perturbation as same as in (38). For the sake of the readability, cost models will be derived assuming $\mathbf{X}_i \in SE_3(3)$ with a single contact $\mathbf{d}_i$. Since the dynamics act identically for all feet in contact, the developed cost models would be easily augmented to multicontact states without yielding any loss of generality. Now, we consider the sensor measurements $\mathbf{Z}_i$ of the system

$$\mathbf{Z}_i := [\tilde{\omega}_i^T, \tilde{\mathbf{a}}_i^T, \tilde{\mathbf{q}}_i^T]^T \qquad (41)$$

where $\tilde{\omega}_i \in \mathbb{R}^3$, $\tilde{\mathbf{a}}_i \in \mathbb{R}^3$, $\tilde{\mathbf{q}}_i := [(^1\tilde{\mathbf{q}}_i)^T, \ldots, (^N\tilde{\mathbf{q}}_i)^T]^T \in \mathbb{R}^{3N}$ is the gyroscope, accelerometer, and joint encoder measurements, respectively, for N-legged robots, and $^{i_{\text{leg}}}\tilde{\mathbf{q}}_i \in \mathbb{R}^3$ is the i_{leg}th leg's encoder measurement. Following Section III-C with the given state definitions and the sensor measurements, a MAP problem representing the state-estimation problem can be defined as follows:

$$\mathbf{X}_{0:n}^*, \mathbf{x}_{0:n}^* = \underset{\mathbf{X}_{0:n}, \mathbf{x}_{0:n}}{\text{argmax}} \ p(\mathbf{X}_{0:n}, \mathbf{x}_{0:n} \mid \mathbf{Z}_{0:n})$$

$$= \underset{\mathbf{X}_{0:n}, \mathbf{x}_{0:n}}{\text{argmax}} \ p(\mathbf{X}_{0:n}, \mathbf{x}_{0:n}) p(\mathbf{Z}_{0:n} | \mathbf{X}_{0:n}, \mathbf{x}_{0:n}). \qquad (42)$$

In this article, we assume that the above posterior probability distribution is decomposed into *prior* distribution, *propagation* distribution, *observation* distribution, and *loop* distribution (included for the CL method, which will be further explained in Section V) as follows:

$$\mathbf{X}_{0:n}^*, \mathbf{x}_{0:n}^*$$

$$= \underset{\mathbf{X}_{0:n}, \mathbf{x}_{0:n}}{\text{argmax}} \ \underbrace{p(\mathbf{X}_0, \mathbf{x}_0)}_{\text{Prior}} \underbrace{\prod_{i=0}^{n-1} p(\mathbf{X}_{i+1}, \mathbf{x}_{i+1} \mid \mathbf{X}_i, \mathbf{x}_i, \mathbf{Z}_i)}_{\text{Propagation}}$$

$$\underbrace{\prod_{i=0}^{n} p(\mathbf{Z}_i \mid \mathbf{X}_i, \mathbf{x}_i)}_{\text{Observation}} \underbrace{\prod_{a,b \in \mathscr{L}} p(\mathbf{Z}_{a:b} \mid \mathbf{X}_a, \mathbf{x}_a, \mathbf{X}_b, \mathbf{x}_b)}_{\text{Loop}} \qquad (43)$$

where $\mathbf{Z}_i$ is the sensor observation at ith timestep, and $\mathscr{L}$ stands for the set of the long-term observations whereas a and b stand for the start timestep and the end timestep of the long-term observation, respectively. Similarly as (34), (43) is further transformed into a nonlinear least-square problem

$$\mathbf{X}_{0:n}^*, \mathbf{x}_{0:n}^* = \underset{\mathbf{X}_{0:n}, \mathbf{x}_{0:n}}{\text{argmin}} \left(\|\mathbf{r}_{\text{Pri}}\|_{\boldsymbol{\Sigma}_{\text{Pri}}}^2 + \sum_{i=0}^{n-1} \|\mathbf{r}_{\text{Prop}_i}\|_{\boldsymbol{\Sigma}_{\text{Prop}_i}}^2 \right.$$

$$\left. + \sum_{i=0}^{n} \|\mathbf{r}_{\text{Obs}_i}\|_{\boldsymbol{\Sigma}_{\text{Obs}_i}}^2 + \sum_{a,b \in \mathscr{L}} \|\mathbf{r}_{\text{Loop}_{a,b}}\|_{\boldsymbol{\Sigma}_{\text{Loop}_{a,b}}}^2 \right) \qquad (44)$$

where $\mathbf{r}_{\text{Pri}}$ and $\boldsymbol{\Sigma}_{\text{Pri}}$ is the corresponding residual function and covariance matrix of the prior distribution, $\mathbf{r}_{\text{Prop}_i}$ and $\boldsymbol{\Sigma}_{\text{Prop}_i}$ is the corresponding residual function and covariance matrix of the propagation distribution, $\mathbf{r}_{\text{Obs}_i}$ and $\boldsymbol{\Sigma}_{\text{Obs}_i}$ is the corresponding residual function and covariance matrix of the observation distribution, and $\mathbf{r}_{\text{Loop}_{a,b}}$ and $\boldsymbol{\Sigma}_{\text{Loop}_{a,b}}$ is the corresponding residual function and covariance matrix of the loop distribution. Now, the above problem can be rewritten by giving the perturbation $\mathbf{e}_{0:n}$ defined in (40) at the current operating point

$$\mathbf{e}_{0:n}^* = \text{argmin}(\|\bar{\mathbf{r}}_{\text{Pri}} - \mathbf{J}_{\text{Pri}}\mathbf{e}_{0:n}\|_{\boldsymbol{\Sigma}_{\text{Pri}}}^2$$

$$+ \sum_{i=0}^{n-1} \|\bar{\mathbf{r}}_{\text{Prop}_i} - \mathbf{J}_{\text{Prop}_i}\mathbf{e}_{0:n}\|_{\boldsymbol{\Sigma}_{\text{Prop}_i}}^2$$

$$+ \sum_{i=0}^{n} \|\bar{\mathbf{r}}_{\text{Obs}_i} - \mathbf{J}_{\text{Obs}_i}\mathbf{e}_{0:n}\|_{\boldsymbol{\Sigma}_{\text{Obs}_i}}^2$$

$$+ \sum_{a,b \in \mathscr{L}} \|\bar{\mathbf{r}}_{\text{Loop}_{a,b}} - \mathbf{J}_{\text{Loop}_{a,b}}\mathbf{e}_{0:n}\|_{\boldsymbol{\Sigma}_{\text{Loop}_{a,b}}}^2) \qquad (45)$$

whereas the update at each iteration is done as follows:

$$\mathbf{X}_i^* \leftarrow \text{Exp}(\xi_i^*)\bar{\mathbf{X}}_i, \ \mathbf{x}_i^* \leftarrow \bar{\mathbf{x}}_i + \zeta_i^*. \tag{46}$$

In the next section, we will precisely describe the cost functions that compose the given problem.

B. Detailed Derivation of Cost Functions

1) Prior Cost: The prior cost model sets the prior belief of the state-estimation system when the estimator is initialized. We assume that the initial state is perturbed by the zero-mean Gaussian noise $\mathbf{w}^{\text{Pri},M} = [(\mathbf{w}^{\text{Pri}_\phi})^T, (\mathbf{w}^{\text{Pri}_\mathbf{v}})^T, (\mathbf{w}^{\text{Pri}_\mathbf{P}})^T]^T$ on the lifted tangential space of the manifold variable $\mathbf{X}_0$, and $\mathbf{w}^{\text{Pri},v} = [(\mathbf{w}^{\text{Pri}_{\mathbf{b}^\omega}})^T, (\mathbf{w}^{\text{Pri}_{\mathbf{b}^a}})^T]^T$ of the variable $\mathbf{x}_0$ in vector spaces

$$\underbrace{\left[\ \mathbb{I}_5 \quad \mathbf{0}_{5,1}\ \right]}_{\mathbf{M}_{\text{Pri}}}\mathbf{X}_0\mathbf{M}_{\text{Pri}}^T = \begin{bmatrix} \mathbf{R}_0 & \mathbf{v}_0 & \mathbf{p}_0 \\ \mathbf{0}_{1\times 3} & 1 & 0 \\ \mathbf{0}_{1\times 3} & 0 & 1 \end{bmatrix}$$

$$= \text{Exp}(\mathbf{w}^{\text{Pri},M})\mathbf{X}_{\text{Pri}}$$

$$\mathbf{x}_0 = \mathbf{x}_{\text{Pri}} + \mathbf{w}^{\text{Pri},v}. \tag{47}$$

$\mathbf{X}_{\text{Pri}} \in \mathbb{R}^{5\times 5}$ is the prior belief of $\mathbf{R}_0$, $\mathbf{v}_0$, and $\mathbf{p}_0$, and $\mathbf{M}_{\text{Pri}}$ is the auxiliary block operator matrix to convert $\mathbf{X}_0 \in \mathbb{R}^{6\times 6}$ to $\mathbb{R}^{5\times 5}$ without $\mathbf{d}_0$. $\mathbf{x}_{\text{Pri}} \in \mathbb{R}^6$ is the prior belief of bias. We do not set prior belief on $\mathbf{d}_0$, because $\mathbf{d}_0$ is initialized later by the observation factor at the initial timestep through the forward kinematics relationship. The left-hand side of the prior cost model for the manifold variable can be further expanded by replacing $\mathbf{X}_0$ with $\text{Exp}(\xi_0)\bar{\mathbf{X}}_0$ as follows:

$$\mathbf{M}_{\text{Pri}}(\text{Exp}(\xi_0)\bar{\mathbf{X}}_0)\mathbf{M}_{\text{Pri}}^T = \text{Exp}(\mathbf{w}^{\text{Pri},M})\mathbf{X}_{\text{Pri}}. \tag{48}$$

Note that by multiplying $\mathbf{M}_{\text{Pri}}$ twice, the left-hand side of the equation does not contain any information of $\mathbf{d}_0$ and ξ_0^d. This means that the left-hand side of the equation is equivalent to

$$\text{Exp}(\mathbf{M}_0\xi_0)(\mathbf{M}_{\text{Pri}}\bar{\mathbf{X}}_0\mathbf{M}_{\text{Pri}}^T) = \text{Exp}(\mathbf{w}^{\text{Pri},M})\mathbf{X}_{\text{Pri}}.$$
$$\text{where } \mathbf{M}_0 = [\ \mathbb{I}_9 \quad \mathbf{0}_{9,3}\ ]. \tag{49}$$

Next, moving $\mathbf{X}_{\text{Pri}}$ to the left-hand side and introducing $\xi_{\text{Pri}} \equiv \text{Log}(\mathbf{X}_{\text{Pri}}(\mathbf{M}_{\text{Pri}}\bar{\mathbf{X}}_0\mathbf{M}_{\text{Pri}}^T)^{-1})$ which stands for the distance between the prior belief and the current operating point, makes the equation as

$$\text{Exp}(\mathbf{M}_0\xi_0)\text{Exp}(-\xi_{\text{Pri}}) = \text{Exp}(\mathbf{w}^{\text{Pri},M}). \tag{50}$$

Finally, taking logarithmic mapping on both sides of the equation while using the BCH formula (16), the resultant equation for the prior of the manifold variable becomes

$$\mathbf{w}^{\text{Pri},M} = \mathbf{M}_0\xi_0 - \xi_{\text{Pri}}. \tag{51}$$

We neglect the Jacobian term in the BCH formula since both $\mathbf{M}_0\xi_0$ and ξ_{Pri} are infinitesimal. Similarly, we can express the prior belief equation of the vector space variable $\mathbf{x}_0$ with ζ_0 by introducing a new variable $\zeta_{\text{Pri}} \equiv \mathbf{x}_{\text{Pri}} - \bar{\mathbf{x}}_0$

$$\mathbf{w}^{\text{Pri},v} = \zeta_0 - \zeta_{\text{Pri}}. \tag{52}$$

Concatenating this with (51) makes the prior cost model of the concatenated perturbation $\mathbf{e}$ and noise $\mathbf{w}^{\text{Pri}} \equiv$ $[(\mathbf{w}^{\text{Pri},M})^T, (\mathbf{w}^{\text{Pri},v})^T]^T$

$$\mathbf{w}^{\text{Pri}} = \begin{bmatrix} \mathbf{M}_0 & \mathbf{0}_{9,6} \\ \mathbf{0}_{6,9} & \mathbb{I}_6 \end{bmatrix}\mathbf{e}_0 - \mathbf{e}_{\text{Pri}}. \tag{53}$$

Finally, the residual function $\mathbf{r}_{\text{Pri}}$, Jacobian $\mathbf{J}_{\text{Pri}}$, and its corresponding covariance matrix $\mathbf{\Sigma}_{\text{Pri}}$ can now be summarized as follows:

$$\mathbf{r}_{\text{Pri}} = - \left[\text{Log}(\mathbf{X}_{\text{Pri}}(\mathbf{M}_{\text{Pri}}\bar{\mathbf{X}}_0\mathbf{M}_{\text{Pri}}^T)^{-1})^T, (\mathbf{x}_{\text{Pri}} - \bar{\mathbf{x}}_0)^T\right]^T$$

$$\mathbf{J}_{\text{Pri}} = \begin{bmatrix} \mathbf{M}_0 & \mathbf{0}_{9,6} \\ \mathbf{0}_{6,9} & \mathbb{I}_6 \end{bmatrix}, \ \mathbf{\Sigma}_{\text{Pri}} = \text{Cov}(\mathbf{w}^{\text{Pri}}). \tag{54}$$

2) Propagation Cost: Propagation cost is derived from the system dynamics equation of the states in the continuous domain. The modeling incorporates the IMU measurements by integrating acceleration to velocity, velocity to position, and angular velocity to rotation. Simultaneously, the stable foot contact assumption ensures that the contact foot position remains stable over time, leading to the assumption that the foot's velocity is zero [32]. It is also important to consider the potential noise of the system. Specifically, the IMU measurements are assumed to be corrupted by slowly changing bias and zero-mean white Gaussian sensor noise

$$\tilde{\omega}_t = \omega_t + \mathbf{b}_t^\omega + \mathbf{w}^\omega, \quad \mathbf{w}^\omega \sim \mathcal{N}(\mathbf{0}_{3,1}, \Sigma^\omega)$$
$$\tilde{\mathbf{a}}_t = \mathbf{a}_t + \mathbf{b}_t^a + \mathbf{w}^a, \quad \mathbf{w}^a \sim \mathcal{N}(\mathbf{0}_{3,1}, \Sigma^a). \tag{55}$$

where $\tilde{\omega}_t, \tilde{\mathbf{a}}_t$ is the gyroscope and the accelerometer measurement, and $\omega_t, \mathbf{a}_t$ is the true angular velocity and acceleration, respectively, both in local-coordinate. In the case of the stable contact foot assumption, the velocity of the foot is assumed to be zero, whereas additional Gaussian noise $\mathbf{w}^c$ is added to handle the possible violation of the assumption. Now, based on the IMU modeling and the stable contact foot assumption, we can model the propagation of the states as follows:

$$\frac{\text{d}}{\text{d}t}\mathbf{R}_t = \mathbf{R}_t (\tilde{\omega}_t - \mathbf{b}_t^\omega - \mathbf{w}^\omega)^\wedge, \ \frac{\text{d}}{\text{d}t}\mathbf{p}_t = \mathbf{v}_t$$

$$\frac{\text{d}}{\text{d}t}\mathbf{v}_t = \mathbf{R}_t (\tilde{\mathbf{a}}_t - \mathbf{b}_t^a - \mathbf{w}^a) + \mathbf{g}, \ \frac{\text{d}}{\text{d}t}\mathbf{d}_t = \mathbf{R}_t\mathbf{w}^c$$

$$\frac{\text{d}}{\text{d}t}\mathbf{b}_t^\omega = \mathbf{w}^{b^\omega}, \quad \frac{\text{d}}{\text{d}t}\mathbf{b}_t^a = \mathbf{w}^{b^a} \tag{56}$$

where $\mathbf{g}$ is the gravity vector. $\mathbf{w}^\omega, \mathbf{w}^a, \mathbf{w}^c, \mathbf{w}^{b^\omega}$, and $\mathbf{w}^{b^a}$ are the zero-mean Gaussian noises associated with each process. We will denote the noise-free propagation functions of the manifold state variable and vector space state variable as $f_M(\cdot)$ and $f_v(\cdot)$, respectively, where the subscript M denotes the dynamics of the $SE_3(3)$ manifold space variables and v denotes for the dynamics of the $\mathbb{R}^6$ vector space variables. Writing explicitly, $f_M(\cdot)$ and $f_v(\cdot)$ are

$$f_M(\mathbf{X}_t) = \begin{bmatrix} \mathbf{R}_t (\tilde{\omega}_t - \mathbf{b}_t^\omega)^\wedge & \mathbf{R}_t (\tilde{\mathbf{a}}_t - \mathbf{b}_t^a) + \mathbf{g} & \mathbf{v}_t & \mathbf{0}_{3\times 1} \\ \mathbf{0}_{3\times 3} & & & \mathbb{I}_3 \end{bmatrix}$$

$$f_v(\mathbf{x}_t) = \begin{bmatrix} \mathbf{0}_{3\times 1} \\ \mathbf{0}_{3\times 1} \end{bmatrix}. \tag{57}$$

For these propagation functions, Hartley et al. [32] came up with the nearly log-linear error propagation equation in the

continuous domain by utilizing (25)

$$\frac{d}{dt}\begin{bmatrix}\xi_t\\\zeta_t\end{bmatrix} = \mathbf{A}_t\begin{bmatrix}\xi_t\\\zeta_t\end{bmatrix} + \begin{bmatrix}\mathrm{Ad}_{\bar{\mathbf{X}}_t} & \mathbf{0}_{12,6}\\\mathbf{0}_{6,12} & \mathbb{I}_6\end{bmatrix}{}^c\mathbf{w}^{\mathrm{Prop}}, \quad (58)$$

where ${}^c\mathbf{w}^{\mathrm{Prop}} = [(\mathbf{w}^\omega)^T, (\mathbf{w}^\mathrm{a})^T, \mathbf{0}_{1,3}, (\mathbf{w}^\mathrm{c})^T, (\mathbf{w}^{\mathbf{b}^\omega})^T, (\mathbf{w}^{\mathbf{b}^\mathrm{a}})^T]^T \in \mathbb{R}^{18}$ and $\mathbf{A}_t$ is:

$$\mathbf{A}_t = \begin{bmatrix}\mathbf{0}_{3,3} & \mathbf{0}_{3,3} & \mathbf{0}_{3,3} & \mathbf{0}_{3,3} & -\overline{\mathbf{R}}_t & \mathbf{0}_{3,3}\\(\mathbf{g})^\wedge & \mathbf{0}_{3,3} & \mathbf{0}_{3,3} & \mathbf{0}_{3,3} & -(\overline{\mathbf{v}}_t)^\wedge\overline{\mathbf{R}}_t & -\overline{\mathbf{R}}_t\\\mathbf{0}_{3,3} & \mathbb{I}_3 & \mathbf{0}_{3,3} & \mathbf{0}_{3,3} & -(\overline{\mathbf{p}}_t)^\wedge\overline{\mathbf{R}}_t & \mathbf{0}_{3,3}\\\mathbf{0}_{3,3} & \mathbf{0}_{3,3} & \mathbf{0}_{3,3} & \mathbf{0}_{3,3} & -(\overline{\mathbf{d}}_t)^\wedge\overline{\mathbf{R}}_t & \mathbf{0}_{3,3}\\\mathbf{0}_{3,3} & \mathbf{0}_{3,3} & \mathbf{0}_{3,3} & \mathbf{0}_{3,3} & \mathbf{0}_{3,3} & \mathbf{0}_{3,3}\\\mathbf{0}_{3,3} & \mathbf{0}_{3,3} & \mathbf{0}_{3,3} & \mathbf{0}_{3,3} & \mathbf{0}_{3,3} & \mathbf{0}_{3,3}\end{bmatrix}. \quad (59)$$

To formulate the smoother, we discretize (58) and rearrange it to the form of residual function. First, we make discrete domain propagation functions $f_M^d(\cdot)$ and $f_v^d(\cdot)$ by discretizing $f_M(\cdot)$ and $f_v(\cdot)$. Whereas various discretization methods could take place, we adopted the following discretization:

$$f_M^d(\mathbf{X}_i) = \begin{bmatrix}\mathbf{R}_i\mathrm{Exp}((\tilde{\omega}_i - \mathbf{b}_i^\omega)\Delta t) & \mathbf{v}_i^d & \mathbf{p}_i^d & \mathbf{d}_i\\\mathbf{0}_{1,3} & 1 & 0 & 0\\\mathbf{0}_{1,3} & 0 & 1 & 0\\\mathbf{0}_{1,3} & 0 & 0 & 1\end{bmatrix}$$

$$f_v^d(\mathbf{x}_i) = \begin{bmatrix}\mathbf{b}_i^\omega\\\mathbf{b}_i^a\end{bmatrix} \quad (60)$$

where $\mathbf{v}_i^d = \mathbf{v}_i + \mathbf{R}_i(\tilde{\mathbf{a}}_i - \mathbf{b}_i^\mathrm{a})\Delta t + \mathbf{g}\Delta t$ and $\mathbf{p}_i^d = \mathbf{p}_i + \mathbf{v}_i\Delta t + \frac{1}{2}\mathbf{R}_i(\tilde{\mathbf{a}}_i - \mathbf{b}_i^\mathrm{a})(\Delta t)^2 + \frac{1}{2}\mathbf{g}(\Delta t)^2$. Then, we introduce new auxiliary error variables ξ_{i+1}^f and ζ_{i+1}^f

$$\xi_{i+1}^f = \mathrm{Log}(\mathbf{X}_{i+1}f_M^d(\bar{\mathbf{X}}_i)^{-1}), \quad \zeta_{i+1}^f = \mathbf{x}_{i+1} - f_v^d(\bar{\mathbf{x}}_i). \quad (61)$$

The auxiliary variables are needed for the discretization of (58) since the equation uses the log-linear property. As illustrated in Fig. 4, the log-linear property is applicable only to the errors defined by two trajectories propagated by the same propagation function. Therefore, the discretized log-linear property of ξ_i is satisfied only with ξ_{i+1}^f but not ξ_{i+1}, which represents the error of the current operating point $\bar{\mathbf{X}}_{i+1}$. With this nature of the log-linear property, we can discretize (58) using the forward Euler method

$$\begin{bmatrix}\xi_{i+1}^f\\\zeta_{i+1}^f\end{bmatrix} = (\mathbb{I}_{18} + \mathbf{A}_i\Delta t)\begin{bmatrix}\xi_i\\\zeta_i\end{bmatrix}\begin{bmatrix}+\mathrm{Ad}_{\bar{\mathbf{X}}_i} & \mathbf{0}_{12,6}\\\mathbf{0}_{6,12} & \mathbb{I}_6\end{bmatrix}\Delta t\,{}^d\mathbf{w}^{Prop} \quad (62)$$

where Δt is the discretizing time interval. Inspired by [10], we use the second-order IMU discretization model $\mathbf{p}_{i+1} = \mathbf{p}_i + \mathbf{v}_i\Delta t + 0.5\mathbf{a}_i\Delta t^2$ for position propagation instead of exploiting only the first derivative, as the second derivative $\mathbf{a}$ is available. As a result, we use an uncertainty model ${}^d\mathbf{w}^{\mathrm{Prop}} = [(\mathbf{w}^\omega)^T, (\mathbf{w}^\mathrm{a})^T, (\mathbf{w}^\mathrm{a}\Delta t)^T, (\mathbf{w}^\mathrm{c})^T, (\mathbf{w}^{\mathbf{b}^\omega})^T, (\mathbf{w}^{\mathbf{b}^\mathrm{a}})^T]^T$, where we add $\mathbf{w}^\mathrm{a}\Delta t$ on ${}^c\mathbf{w}^{\mathrm{Prop}}$ to deal with the noise of accelerometer intervening to the discrete position propagation. Now, we further decompose the left-hand side of the discretized (62), so that it comprises the perturbation variable $[\xi_{i+1}^T, \zeta_{i+1}^T]^T$:

$$\begin{bmatrix}\xi_{i+1}^f\\\zeta_{i+1}^f\end{bmatrix} = \begin{bmatrix}\mathrm{Log}(\mathbf{X}_{i+1}f_M^d(\bar{\mathbf{X}}_i)^{-1})\\\mathbf{x}_{i+1} - f_v^d(\bar{\mathbf{x}}_i)\end{bmatrix}$$
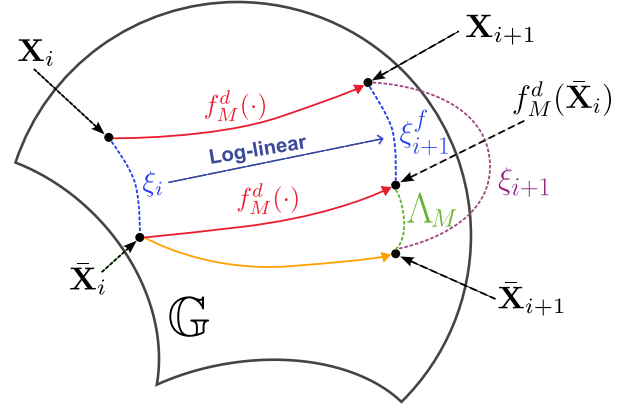

Fig. 4. Graphical illustration of discretization of continuous log-linear property of a manifold variable. A vector space variable is not presented for readability. Propagation processes of manifold variables are presented by bold-lined arrows, where the system propagation $f_M^d(\cdot)$ is colored especially by red. Log invariant errors between manifold variables are expressed by dotted lines, where the log invariant errors that satisfy the log-linear property are colored blue. Note that where the log-linear property of the errors is applied is consistent with where $f_M^d(\cdot)$ is applied on the manifold.

$$= \begin{bmatrix}\mathrm{Log}(\mathbf{X}_{i+1}\bar{\mathbf{X}}_{i+1}^{-1}\bar{\mathbf{X}}_{i+1}f_M^d(\bar{\mathbf{X}}_i)^{-1})\\\mathbf{x}_{i+1} - \bar{\mathbf{x}}_{i+1} + \bar{\mathbf{x}}_{i+1} - f_v^d(\bar{\mathbf{x}}_i)\end{bmatrix}$$

$$= \begin{bmatrix}\mathrm{Log}(\mathrm{Exp}(\xi_{i+1})\mathrm{Exp}(-\Lambda_M))\\\zeta_{i+1} - \Lambda_v\end{bmatrix} \approx \begin{bmatrix}\xi_{i+1}\\\zeta_{i+1}\end{bmatrix} - \Lambda \quad (63)$$

where Λ_M, Λ_v are $\mathrm{Log}(f_M^d(\bar{\mathbf{X}}_i)\bar{\mathbf{X}}_{i+1}^{-1})$, $(f_v^d(\bar{\mathbf{x}}_i) - \bar{\mathbf{x}}_{i+1})$, respectively, and Λ is $[\Lambda_M^T, \Lambda_v^T]^T$. In the last operation of (63), we applied the BCH formula (16) and assumed that the Λ_M is small, neglecting the Jacobian of Λ_M. Now, the propagation residual function $\mathbf{r}_{\mathrm{Prop}_i}$, Jacobians $\mathbf{J}_{\mathrm{Prop}_{i+1}}$, $\mathbf{J}_{\mathrm{Prop}_i}$, and corresponding covariance matrix $\boldsymbol{\Sigma}_{\mathrm{Prop}_i}$ can be derived

$$\mathbf{r}_{\mathrm{Prop}_i} = -\begin{bmatrix}\mathrm{Log}(f_M^d(\bar{\mathbf{X}}_i)\bar{\mathbf{X}}_{i+1}^{-1})\\f_v^d(\bar{\mathbf{x}}_i) - \bar{\mathbf{x}}_{i+1}\end{bmatrix}$$

$$\mathbf{J}_{\mathrm{Prop}_i} = -(\mathbb{I}_{18} + \mathbf{A}_i\Delta t), \quad \mathbf{J}_{\mathrm{Prop}_{i+1}} = \mathbb{I}_{18}$$

$$\boldsymbol{\Sigma}_{\mathrm{Prop}_i} = \mathbf{A}_{\mathrm{Prop}_i}\mathrm{Cov}({}^d\mathbf{w}^{\mathrm{Prop}})\mathbf{A}_{\mathrm{Prop}_i}^T$$

$$\mathbf{A}_{\mathrm{Prop}_i} = \begin{bmatrix}\mathrm{Ad}_{\bar{\mathbf{X}}_i} & \mathbf{0}_{12,6}\\\mathbf{0}_{6,12} & \mathbb{I}_6\end{bmatrix}\Delta t. \quad (64)$$

3) Observation Cost: The observation cost function is established by the encoder measurements that relate body pose and foot position. Encoder measurements are assumed to be corrupted by zero-mean white Gaussian

$$\tilde{\mathbf{q}}_i = \mathbf{q}_i + \mathbf{w}^\mathbf{q}, \quad \mathbf{w}^\mathbf{q} \sim \mathcal{N}(\mathbf{0}_{3,1}, \Sigma^\mathbf{q}) \quad (65)$$

where $\mathbf{q}_i$ are the true joint positions and $\mathbf{w}^\mathbf{q}$ is the encoder sensor noise. The encoder measurements are then exploited to formulate a forward kinematics relationship.

$$\mathbf{d}_i = \mathbf{R}_i(fk(\tilde{\mathbf{q}}_i) - \mathbf{J}_{fk}(\tilde{\mathbf{q}}_i)\mathbf{w}^\mathbf{q}) + \mathbf{p}_i \quad (66)$$

where $fk(\tilde{\mathbf{q}}_i)$ and $\mathbf{J}_{fk}(\tilde{\mathbf{q}}_i)$ are the forward kinematics of the robot and the Jacobian of the forward kinematics function. Expressing the relationship analogous to (28), the observation

cost model becomes

$$
\underbrace{\begin{bmatrix} fk(\tilde{\mathbf{q}}_i) \\ 0 \\ 1 \\ -1 \end{bmatrix}}_{\mathbf{y}_i}
$$

$$
= \underbrace{\begin{bmatrix} \mathbf{R}_i^\top & -\mathbf{R}_i^\top \mathbf{v}_i & -\mathbf{R}_i^\top \mathbf{p}_i & -\mathbf{R}_i^\top \mathbf{d}_i \\ \mathbf{0}_{1,3} & 1 & 0 & 0 \\ \mathbf{0}_{1,3} & 0 & 1 & 0 \\ \mathbf{0}_{1,3} & 0 & 0 & 1 \end{bmatrix}}_{\mathbf{X}_i^{-1}} \underbrace{\begin{bmatrix} \mathbf{0}_{3,1} \\ 0 \\ 1 \\ -1 \end{bmatrix}}_{\mathbf{s}}
$$

$$
+ \underbrace{\begin{bmatrix} \mathbf{J}_{fk}(\tilde{\mathbf{q}}_i)\mathbf{w}^{\mathbf{q}} \\ 0 \\ 0 \\ 0 \end{bmatrix}}_{\mathbf{w}_i^{\mathrm{obs}}}. \tag{67}
$$

Now left-multiplying $\bar{\mathbf{X}}_i$ on both sides, the equation becomes

$$
\bar{\mathbf{X}}_i \mathbf{y}_i = \mathrm{Exp}(-\xi_i)\mathbf{s} + \bar{\mathbf{X}}_i \mathbf{w}_i^{\mathrm{Obs}}. \tag{68}
$$

Presuming that ξ_i is small, we can linearize the observation model by applying the first-order approximation of the exponential map (17)

$$
\bar{\mathbf{X}}_i \mathbf{w}_i^{\mathrm{Obs}} = \bar{\mathbf{X}}_i \mathbf{y}_i - \mathbf{s} + \mathbf{s}^\odot \xi_i \tag{69}
$$

where the linearized model is reorganized by using (18). Now, residual function $\mathbf{r}_{\mathrm{Obs}_i}$, Jacobian $\mathbf{J}_{\mathrm{Obs}_i}$, and its corresponding covariance matrix $\boldsymbol{\Sigma}_{\mathrm{Obs}_i}$ can be derived

$$
\mathbf{r}_{\mathrm{Obs}_i} = \bar{\mathbf{X}}_i \mathbf{y}_i - \mathbf{s}, \quad \mathbf{J}_{\mathrm{Obs}_i} = \begin{bmatrix} \mathbf{s}^\odot & \mathbf{0}_{6,6} \end{bmatrix}
$$

$$
\boldsymbol{\Sigma}_{\mathrm{Obs}_i} = \bar{\mathbf{X}}_i \, \mathrm{Cov}(\mathbf{w}_i^{\mathrm{Obs}}) \bar{\mathbf{X}}_i^T. \tag{70}
$$

4) Marginalized Prior Cost: While the cost models for batch estimation are summarized above, restricting the growing size of the problem is necessary for practical implementation. We alleviate this problem by adopting a commonly used marginalization method from [55]

$$
\mathbf{w}_i^{\mathrm{Marg}} = \mathbf{r}^* + (\mathbf{e}_{\mathrm{Old}} - \mathbf{e}_{\mathrm{Marg}})
$$

where $\quad \mathbf{w}_i^{\mathrm{Marg}} \sim \mathcal{N}(\mathbf{0}_{3,1}, (\mathbf{H}^*)^{-1}), \ \mathbf{r}^* = (\mathbf{H}^*)^{-1}\mathbf{z}^*. \tag{71}$

Here, $\mathbf{H}^*$ and $\mathbf{z}^*$ are the Schur complement of the Hessian and gradient constructed right before the marginalization. The detailed derivation is explained in [55] and not repeated here. $\mathbf{e}_{\mathrm{Old}}$ denotes the perturbation at the oldest timestep in the current window, and $\mathbf{e}_{\mathrm{Marg}} = \begin{bmatrix} \mathrm{Log}(\mathbf{X}_{\mathrm{Marg}}\bar{\mathbf{X}}_{\mathrm{Old}}^{-1})^T, (\mathbf{x}_{\mathrm{Marg}} - \bar{\mathbf{x}}_{\mathrm{Old}})^T \end{bmatrix}^T$ denotes the difference between the operating points at the oldest timestep in the window $\bar{\mathbf{X}}_{\mathrm{Old}}, \bar{\mathbf{x}}_{\mathrm{Old}}$ and the marginalized prior states $\mathbf{X}_{\mathrm{Marg}}, \mathbf{x}_{\mathrm{Marg}}$, which are initialized right after the marginalization. Finally, residual function $\mathbf{r}_{\mathrm{Marg}}$, Jacobian $\mathbf{J}_{\mathrm{Old}}$, and its corresponding covariance matrix $\boldsymbol{\Sigma}_{\mathrm{Marg}}$ can be derived

$$
\mathbf{r}_{\mathrm{Marg}} = \mathbf{r}^* - \begin{bmatrix} \mathrm{Log}(\mathbf{X}_{\mathrm{Marg}}\bar{\mathbf{X}}_{\mathrm{Old}}^{-1})^T, (\mathbf{x}_{\mathrm{Marg}} - \bar{\mathbf{x}}_{\mathrm{Old}})^T \end{bmatrix}^T
$$

$$
\mathbf{J}_{\mathrm{Old}} = \mathbb{I}_{18}, \ \boldsymbol{\Sigma}_{\mathrm{Marg}} = (\mathbf{H}^*)^{-1}. \tag{72}
$$

In summary, the Jacobians of the cost functions described in the previous section are state-independent, except for the Jacobian

about IMU biases in the propagation cost. The state-independent Jacobians can relieve the iterative optimization process and improve the estimation results, as we will see in Section VII. On the other hand, if the cost functions for optimization are dependent on the state, significant variations in the states resulting from dynamic contact events would distort the convergence characteristics of the optimization and, therefore, the estimation outcomes. While the cost functions required to implement the proposed IS are described in the previous sections, additional methods that can further exploit the static contact assumption can be considered. For this reason, the following section will describe the SR method and CL method, which handle the situations when the static contact assumption is violated and applied with a temporal displacement cost, respectively.

V. LEGGED ROBOT SPECIFIC METHODS

In the case of proprioceptive sensor-only legged robot state estimation, kinematic measurements are utilized on the assumption of the static contact of the foot. For this perspective, the following sections describe the methods to discreetly leverage the static contact foot assumption based on the advantage of smoothing structures.

A. Slip Rejection Method

The static contact assumption of the foot is the key assumption used for utilizing the kinematic measurements of the legged robot. However, this assumption is frequently violated in the real world for various reasons, including slippage of the foot, unstable ground, and late or early contact detection. For this reason, we adopt a method called SR from our previous work [18], which is triggered by thresholding the estimated foot velocity $\bar{\mathbf{v}}_{\mathrm{foot}_i} = \bar{\mathbf{v}}_i + \bar{\mathbf{R}}_i \mathbf{J}_{fk}(\tilde{\mathbf{q}}_i)\dot{\tilde{\mathbf{q}}}_i + \bar{\mathbf{R}}_i(\tilde{\omega}_i - \bar{\mathbf{b}}_i^\omega)^\wedge fk(\tilde{\mathbf{q}}_i)$, where $\dot{\tilde{\mathbf{q}}}_i \in \mathbb{R}^3$ is the joint velocity obtained by numerical differentiation of $\tilde{\mathbf{q}}_i$. The original implementation of the SR method in [18] denies the kinematic measurement if the calculated foot velocity is larger than the preset threshold $\alpha_{\mathbf{v}}$. However, in this work, instead of completely denying the measurement, we increase the uncertainty of the static contact foot assumption as a secure way to implement it in practical applications. The covariance matrix of the large uncertainty is now going to be called *slip covariance* (Σ_s), whereas the original covariance matrix will be called *contact covariance* (Σ_c). These slip covariance and contact covariance are both tuning parameters that users can adjust.

As stated in [18], SR is a useful strategy for both filter-based and smoother-based algorithms to overcome hostile terrains that could significantly degenerate the estimation quality due to the violation of the static contact foot assumption. However, in the case of smoother-based algorithms, there still exists room for additional improvement. Since the foot velocity in the past can be recalculated within its window, the stability of the contact and, hence, the magnitude of the covariance matrix, could be re-evaluated, which would reform the entire estimation results. A graphical illustration of the re-evaluation of the SR method is provided in Fig. 5, where it is possible to identify that the recalculated foot velocity and contact stability can change the formulation of the cost function.
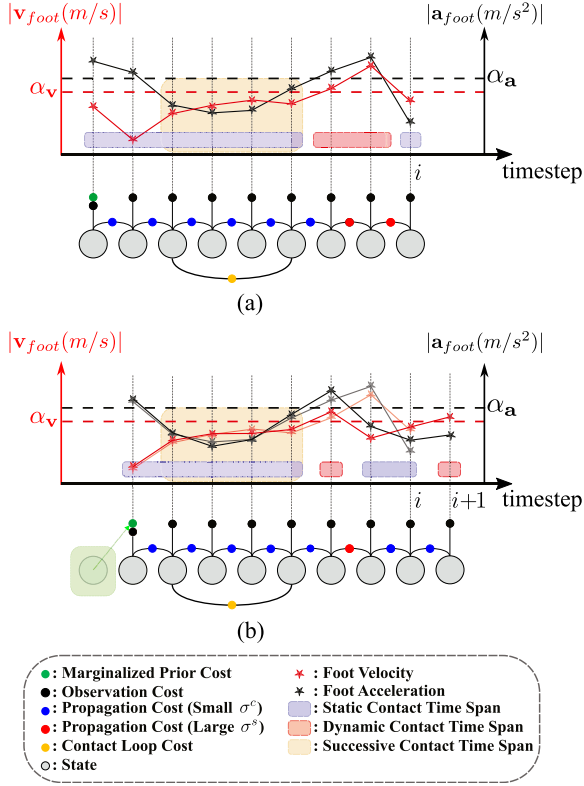
(a)

(b)

Fig. 5. Graphical explanation of the SR method and the CL method, along with the re-evaluation of the foot velocity and acceleration. (a) States and the cost functions evaluated at an exemplary timestep i are represented. (b) Evaluation at timestep $i+1$ is represented. In (a), the propagation cost functions are constructed with the contact covariance (blue) or the slip covariance (red), depending on whether the robot is in a static contact time span (blue-colored span) or a dynamic contact time span (red-colored span). The dynamic contact is assumed when the calculated foot velocity (red stars) exceeds the threshold value $\alpha_{\mathbf{v}}$, whereas the static contact is presumed when the foot velocity is within the threshold. In addition to this SR method, the CL cost (yellow) is formulated between the leftmost and rightmost boundaries of the successive contact time span (yellow-colored span), which is applied when it keeps both foot velocity and acceleration continuously under the threshold values $\alpha_{\mathbf{v}}$ and $\alpha_{\mathbf{a}}$, respectively, at least over two timesteps. In (b), instead of the obsolete foot velocity and acceleration (translucent lines), which were calculated before, the foot velocity and acceleration are recalculated (opaque lines) leveraging the newly given state information at timestep $i+1$ and used to re-evaluate the past contact of the robot. As a result, although the robot was assumed to be in dynamic contact at timestep $i-1$ in (a), now in (b), it is inferred to be in static contact at the same timestep $i-1$.

B. Contact Loop Method

One of the advantages of the smoother framework is that the states within the history window are accessible. Therefore, a measurement model that relates states over distant timestep can be considered for smoothing frameworks, whereas filtering frameworks only deal with the adjacent states. In this point of view, we propose the CL method, which restricts that the foot positions should not be changed over multiple timesteps when there is no slip as follows:

$$\mathbf{r}_{\mathrm{Loop}_{a,b}} = \underbrace{[\ \mathbb{I}_3 \quad \mathbf{0}_{3,3}\ ]}_{\mathbf{M}_{\mathrm{Loop}}}(\bar{\mathbf{X}}_a - \bar{\mathbf{X}}_b)\underbrace{\begin{bmatrix}\mathbf{0}_{3,1}\\0\\0\\1\end{bmatrix}}_{\mathbf{s}} = \bar{\mathbf{d}}_a - \bar{\mathbf{d}}_b$$

$$\mathbf{J}_{\mathrm{Loop}_a} = \mathbf{M}_{\mathrm{Loop}}([\ \mathbf{s}^{\odot} \quad \mathbf{0}_{6,6}\ ])$$
$$\mathbf{J}_{\mathrm{Loop}_b} = -\mathbf{M}_{\mathrm{Loop}}([\ \mathbf{s}^{\odot} \quad \mathbf{0}_{6,6}\ ]), \ \mathbf{\Sigma}_{\mathrm{Loop}_{a,b}} = \Sigma_{\mathrm{cl}}\Delta t \quad (73)$$

where Σ_{cl} is the covariance matrix denoting the user-defined assurance of the CL. Subscript a and b stand for the start and the end of the restricted contact and are determined by the criteria based on the estimated foot motion. The detailed graphical interpretation of this model is given in Fig. 5. A loop is connected over two distant timesteps when none of the absolute values of the estimated foot velocity between the time span is over the threshold $\alpha_{\mathbf{v}}$, and none of the absolute value of the numerically differentiated foot acceleration $\bar{\mathbf{a}}_{\mathrm{foot}_i} = \frac{(\bar{\mathbf{v}}_{\mathrm{foot}_i} - \bar{\mathbf{v}}_{\mathrm{foot}_{i-1}})}{\Delta t}$ is over the threshold $\alpha_{\mathbf{a}}$. The foot acceleration thresholding condition is added to reject the impact situations, such as the sudden vibration of the foot, which might not be captured by the foot velocity criteria.

VI. DIFFERENT STATE DEFINITIONS FOR THE SYSTEM

In addition to the state definition described in the previous section, which includes the position of the foot inside the $\mathrm{SE}_3(3)$ manifold, other kinds of state definitions can be used for legged robot state estimation. In this section, we overview alternative state definitions and their effect on the nonlinearity of the cost function. The state estimators derived from these different state definitions will be compared with our IS later in Section VII.

A. Foot Position in a Vector State

In the previous section, we defined a manifold state variable encompassing the whole state except for bias, which produces state independence. However, we will see in this section that the state definition of foot position outside of the manifold state variable yields the cost models with a partial loss of state independence, as its observation cost function becomes noninvariant. State definition of this system would handle foot position in a vector space variable in $\mathbb{R}^{6+3}$ with the IMU biases, instead of a manifold variable

$$\mathbf{X}_i := \begin{bmatrix}\mathbf{R}_i & \mathbf{v}_i & \mathbf{p}_i\\ \mathbf{0}_{1,3} & 1 & 0\\ \mathbf{0}_{1,3} & 0 & 1\end{bmatrix}, \quad \mathbf{x}_i := \left[(\mathbf{d}_i)^T \quad (\mathbf{b}_i^{\omega})^T \quad (\mathbf{b}_i^{\mathrm{a}})^T\right]^T.$$
$$(74)$$

Similar to Section IV-A, we assume a single contact for the readability and define the error of the system as the concatenation of the right-invariant error of $\bar{\mathbf{X}}$ and the vector space error of $\bar{\mathbf{x}}$

$$\xi_i = -\mathrm{Log}(\mathbf{X}_i\bar{\mathbf{X}}_i^{-1}), \ \zeta_i = \mathbf{x}_i - \bar{\mathbf{x}}_i, \ \mathbf{e}_i = [\xi_i^T, \zeta_i^T]^T \in \mathbb{R}^{18}.$$
$$(75)$$

This state definition results in exactly the same prior cost model as the previously developed model, but the propagation model would be slightly different. We can exploit the log-linear property of the error similar to the continuous propagation model of the proposed IS (58)

$$\frac{d}{dt}\begin{bmatrix}\xi_i\\\zeta_i\end{bmatrix} = \mathbf{A}_{p_i}\begin{bmatrix}\xi_i\\\zeta_i\end{bmatrix} + \begin{bmatrix}\mathrm{Ad}_{\bar{\mathbf{X}}_i} & \mathbf{0}_{9,3} & \mathbf{0}_{9,6}\\ \mathbf{0}_{3,9} & \bar{\mathbf{R}}_i & \mathbf{0}_{3,6}\\ \mathbf{0}_{6,9} & \mathbf{0}_{6,3} & \mathbb{I}_6\end{bmatrix}{}^c\mathbf{w}^{\mathrm{Prop}} \quad (76)$$

with a modified $\mathbf{A}_{p_i}$ matrix:

$$
\mathbf{A}_{p_i} = \begin{bmatrix}
\mathbf{0}_{3,3} & \mathbf{0}_{3,3} & \mathbf{0}_{3,3} & \mathbf{0}_{3,3} & -\overline{\mathbf{R}}_i & \mathbf{0}_{3,3} \\
(\mathbf{g})^\wedge & \mathbf{0}_{3,3} & \mathbf{0}_{3,3} & \mathbf{0}_{3,3} & -(\overline{\mathbf{v}}_i)^\wedge \overline{\mathbf{R}}_i & -\overline{\mathbf{R}}_i \\
\mathbf{0}_{3,3} & \mathbb{I}_3 & \mathbf{0}_{3,3} & \mathbf{0}_{3,3} & -(\overline{\mathbf{p}}_i)^\wedge \overline{\mathbf{R}}_i & \mathbf{0}_{3,3} \\
\mathbf{0}_{3,3} & \mathbf{0}_{3,3} & \mathbf{0}_{3,3} & \mathbf{0}_{3,3} & \mathbf{0}_{3,3} & \mathbf{0}_{3,3} \\
\mathbf{0}_{3,3} & \mathbf{0}_{3,3} & \mathbf{0}_{3,3} & \mathbf{0}_{3,3} & \mathbf{0}_{3,3} & \mathbf{0}_{3,3} \\
\mathbf{0}_{3,3} & \mathbf{0}_{3,3} & \mathbf{0}_{3,3} & \mathbf{0}_{3,3} & \mathbf{0}_{3,3} & \mathbf{0}_{3,3}
\end{bmatrix}.
\tag{77}
$$

Note that the modified $\mathbf{A}_{p_i}$ is state-independent, as much as the $\mathbf{A}_i$ in the previous section, whereas the rotation matrix in the noise term will be canceled out later in the case of diagonal noise covariance. On the other hand, constructing an observation equation using the given state definition is not trivial. Based on the given state definition, the forward kinematics relationship (66) is revisited while modifying the equation based on the $SE_2(3)$ manifold variable $\mathbf{X}_i$ and $\mathbb{R}^{6+3}$ vector space variable $\mathbf{x}_i$. This yields the following encoder observation equation:

$$
\underbrace{\begin{bmatrix}
\mathbb{I}_3 & \mathbf{0}_{3,3} & \mathbf{0}_{3,3} \\
\mathbf{0}_{1,3} & \mathbf{0}_{1,3} & \mathbf{0}_{1,3} \\
\mathbf{0}_{1,3} & \mathbf{0}_{1,3} & \mathbf{0}_{1,3}
\end{bmatrix}}_{\mathbf{M}_p}
\underbrace{\begin{bmatrix}
\mathbf{d}_i \\
\mathbf{b}_i^\omega \\
\mathbf{b}_i^a
\end{bmatrix}}_{\mathbf{x}_i}
+ \underbrace{\begin{bmatrix}
\mathbf{0}_{3,1} \\
0 \\
1
\end{bmatrix}}_{\mathbf{s}_p}
$$

$$
= \underbrace{\begin{bmatrix}
\mathbf{R}_i & \mathbf{v}_i & \mathbf{p}_i \\
\mathbf{0}_{1,3} & 1 & 0 \\
\mathbf{0}_{1,3} & 0 & 1
\end{bmatrix}}_{\mathbf{X}_i}
\underbrace{\begin{bmatrix}
fk(\tilde{\mathbf{q}}_i) \\
0 \\
1
\end{bmatrix}}_{\mathbf{y}_{P_i}}
$$

$$
+ \underbrace{\begin{bmatrix}
\mathbf{R}_i & \mathbf{v}_i & \mathbf{p}_i \\
\mathbf{0}_{1,3} & 1 & 0 \\
\mathbf{0}_{1,3} & 0 & 1
\end{bmatrix}}_{\mathbf{X}_i}
\underbrace{\begin{bmatrix}
\mathbf{J}_{fk}(\tilde{\mathbf{q}}_i)\mathbf{w}^{\mathbf{q}} \\
0 \\
0
\end{bmatrix}}_{\mathbf{w}_{P_i}^{\text{Obs}}}
\tag{78}
$$

which can not take a form of either the left-invariant (27) or the right-invariant (28) observation model since the state $\mathbf{d}_i$ is out of the manifold variable $\mathbf{X}_i$. Applying the first-order approximation and rewriting it with error finally results in the observation cost model

$$
\mathbf{r}_{\text{Obs}_i} = -\bar{\mathbf{X}}_i \mathbf{y}_{p_i} + \mathbf{M}_p \bar{\mathbf{x}}_i + \mathbf{s}_p
$$

$$
\mathbf{J}_{\text{Obs}_i} = \begin{bmatrix}
\bar{\mathbf{R}}_i fk(\tilde{\mathbf{q}}_i) + \bar{\mathbf{p}}_i & \mathbf{0}_{3,3} & -\mathbb{I}_3 & \mathbb{I}_3 & \mathbf{0}_{3,3} & \mathbf{0}_{3,3} \\
\mathbf{0}_{1,3} & \mathbf{0}_{1,3} & \mathbf{0}_{1,3} & \mathbf{0}_{1,3} & \mathbf{0}_{1,3} & \mathbf{0}_{1,3} \\
\mathbf{0}_{1,3} & \mathbf{0}_{1,3} & \mathbf{0}_{1,3} & \mathbf{0}_{1,3} & \mathbf{0}_{1,3} & \mathbf{0}_{1,3}
\end{bmatrix}
$$

$$
\boldsymbol{\Sigma}_{\text{Meas}_i} = \bar{\mathbf{X}}_i \operatorname{Cov}(\mathbf{w}_{p_i}^{\text{Obs}}) \bar{\mathbf{X}}_i^T
\tag{79}
$$

where the detailed derivation is shortened here. We can see that the Jacobian about orientation perturbation is now state-dependent. The smoothing algorithm using this state definition and the cost function is going to be called *pseudo invariant smoother* (PIS).

B. State Without Foot Position, While Only Rotation is in the Manifold

Most of the smoothing algorithms for legged robot state estimation usually define states in vector spaces. A commonly used state definition of fixed-lag smoothers for legged robots defines rotation matrix on the manifold, and others in vector space [16], [17], [18], [23], [27]. In this formulation, the foot
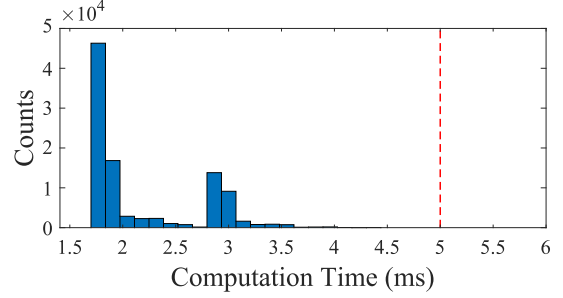

Fig. 6. Histogram of the computation time of the estimation algorithm during the experiments. The red dotted line indicates the computational limit for running the estimator at 200 Hz.

position is not explicitly tracked as a state. According to the definition, the states take the form below:

$$
\mathbf{X}_i := \mathbf{R}_i, \quad \mathbf{x}_i := \begin{bmatrix} (\mathbf{v}_i)^T & (\mathbf{p}_i)^T & (\mathbf{b}_i^\omega)^T & (\mathbf{b}_i^a)^T \end{bmatrix}^T.
\tag{80}
$$

Since only the rotation matrix is defined on the manifold, it is hard to derive state-independent formulations using the group-affine property. The smoothing algorithm for this type of state definition was implemented in [18]. Without loss of generality, comparison against the smoothing algorithm with the orientation state defined as Quaternion is neglected since the local perturbation mapping has a similar hypothesis with NIS.

VII. EXPERIMENTAL RESULTS

In this section, the proposed estimator is verified in real-time experiments and by offline comparison of the performance of the introduced algorithms: IS (proposed), PIS, NIS [18], and InEKF [32]. For a fair comparison, NIS was adopted without *kinematics velocity cost*. Moreover, to be consistent with (62), we used $^d\mathbf{w}^{\text{Prop}}$ as the propagation uncertainty of InEKF, whereas the original InEKF [32] uses $^c\mathbf{w}^{\text{Prop}}$.

At first, we conducted experiments with the quadruped robot *Hound* [3] by controlling the robot based on online feedback of the estimated states from the developed IS. The robot is controlled by the nonlinear model predictive controller based on [2], which runs at 80 Hz. The window of the estimator is set as 15, with a single iteration limit, and turning OFF backtracking to run the estimator online at 200 Hz. The controller and the estimator are run simultaneously on an Intel i7-11700 T 4.6 GHz CPU embedded in the robot. We have recorded the computation time during the running and counted it in histogram Fig. 6, where all logs were reported under the computation time limit of 5 ms. Moreover, thanks to the advantage of invariant frameworks, we could see that the single iterated estimator's performance was retained with less degradation compared with the fully iterated smoother in Table I. This result defends the concern of the single iteration limit and turning OFF the backtracking algorithm for real-time implementation. Consistent with the estimator frequency, IMU and joint encoder measurements were received at 200 Hz. The motion capture device (nine Vicon Vero V2.2 cameras) was used to obtain the ground truth position and orientation at 200 Hz, which could be trusted within 0.5 mm. Also, in order to obtain the ground truth velocity, numerical differentiation of the motion capture data is utilized. The experimental data were obtained from five 120-s-long experiments for the robot walking on unfavorable floor conditions and disturbance. During the experiment, the robot was commanded to walk through the

TABLE I
COMPARISON OF ATE AND 0.6 S RE (MEAN, THE EXCESS OVER THE SMALLEST MEAN VALUE AMONG THE ESTIMATORS, AND STANDARD DEVIATION) BETWEEN IS OF FULL ITERATION AND SINGLE ITERATION

	ATE_{ori}	ATE_{vel}	RE_{ori}	RE_{vel}	RE_{pos}
Full iterations	**0.01984**	**0.19436**	**0.00855**	**0.22716**	**0.05241**
	(0.00266)	(0.01868)	(0.00423)	(0.12812)	(0.04362)
Single iteration	0.01985	0.19555	0.00855	0.22787	0.05392
	(+0.1%)	(+0.6%)	(+0.0%)	(+0.3%)	(+2.9%)
	(0.00266)	(0.01785)	(0.00423)	(0.12867)	(0.04527)

The smallest mean error values are marked in bold.

TABLE II
NOISE PARAMETERS AND THRESHOLDS OF THE ESTIMATOR

Name	Symbol	Value
Gyroscope	σ^{ω}	0.0032 rad/s
Accelerometer	σ^{a}	0.3162 m/s^2
Contact	σ^{c}	0.01 m
Slip	σ^{s}	0.3162 m
Contact Loop	σ^{cl}	0.01 m/s
Encoder	σ^{q}	0.0001 rad
Gyroscope bias	$\sigma^{\mathbf{b}\omega}$	10^{-5} rad/s^2
Accelerometer bias	$\sigma^{\mathbf{b}a}$	$3.162 \cdot 10^{-4}$ m/s^3
Prior orientation	σ^{Pri_ϕ}	10^{-5} rad
Prior velocity	$\sigma^{\text{Pri}_\mathbf{v}}$	10^{-5} m/s
Prior position	$\sigma^{\text{Pri}_\mathbf{p}}$	10^{-5} m
Prior gyroscope bias	$\sigma^{\text{Pri}_{\mathbf{b}\omega}}$	10^{-5} rad/s^2
Prior accelerometer bias	$\sigma^{\text{Pri}_{\mathbf{b}a}}$	10^{-5} m/s^3
Foot velocity threshold	$\alpha_{\mathbf{v}}$	0.3 m/s
Foot acceleration threshold	$\alpha_{\mathbf{a}}$	40 m/s^2

slippery floor, deformable mattress, and piled wood plates with pebble stones. Finally, a strong pulling force was applied to the robot walking on piled wood plates with pebble stones. The detailed environmental setup and the graphical explanation of the experimental procedure are given in Fig. 7. Considering the experimental environment and the noise of the sensors, the tunable noise parameters of the estimator and the thresholds of legged robot-specific methods were empirically chosen as described in Table II. The scalar noise parameters in this table are squared and multiplied by the 3×3 identity matrix to formulate the corresponding covariance matrices.

The ground truth trajectory and the estimation result from one experiment are displayed in Fig. 8, where InEKF was run simultaneously and plotted together for comparison. Note that NIS is removed for comparison in the online experiment since the computation time constraint is not satisfied. The complex trajectory and the large deviation of estimation results are due to the precarious terrains and the disturbance. Featured moments of the scenario are 15–30 s when the robot is on the slippery board and 80–120 s when the robot is on the piled wood plates with pebble stones and disturbed by the strong pulling force. Especially in the 85–95 s, we can identify in the zoomed graph that not only the planar motion but also the orientation and

the velocity of the robot body underwent severe circumstances. Although the estimation was quite noisy in the z-direction, x- and y-directional estimation and orientation estimation tracked the true motion with sufficient reliability. Especially with the proposed estimator, which resulted in a reduced order of estimation bias in velocity, the robot successfully finished the scenario without failure in an extreme environment.

We repeated this experiment five times, and with these five experimental data, we further investigated the proposed estimator's accuracy by the other estimators' various offline evaluations. We allowed smoothing algorithms to take multiple iterations until convergence and turned ON the backtracking line search algorithm for each iteration. Note that the backtracking line search helps the smoother algorithm to find a decent step size and prevent divergence. Smoothers are iterated until convergence within the criteria $\frac{\Delta\text{cost}}{\text{cost}} \leq 0.001$ under the maximal limit of ten iterations, and backtracking was turned ON with the decreasing rate $\gamma = 0.3$ under the maximal backtracking number of $n_{\text{back}} \leq 3$.

The estimation performance is evaluated using absolute trajectory error (ATE) and relative error (RE) from [56] to deliver various properties of the estimators. Within them, RE is adopted with a slight modification in its definition. The new definition addresses the RE not only at the end of the trajectory segment but also during the whole segment. Orientational RE for example, the original definition is $\sqrt{\frac{1}{n_{\text{traj}}} \sum_{i=0}^{n_{\text{traj}}} ||\text{error}_{\phi_i}||^2}$ where n_{traj} is the number of subtrajectories and error_{ϕ_i} is the aligned orientational error at the end of the ith subtrajectory. On the contrary, the newly adopted definition for the orientational RE is $\sqrt{\frac{1}{n_{\text{traj}}} \sum_{i=0}^{n_{\text{traj}}} ||\sqrt{\frac{1}{d_{\text{traj}}} \sum_{j=0}^{d_{\text{traj}}} ||\text{error}_{\phi_j}||^2}||^2}$ where d_{traj} is the length of a subtrajectory and error_{ϕ_j} is the aligned orientational error at the jth timestep in the subtrajectory. This modification is introduced to capture the sudden peaks of errors that frequently occur in legged robot state estimation compared with visual–inertial navigation. In addition, ATE_{pos} is not used, and ATE_{ori} captures only roll and pitch error (without yaw angle error) since global position and yaw angle are not observable in our sensor configuration. Instead, RE_{pos} is calculated, and RE_{ori} captures orientation deviation that includes yaw angle deviation to examine the local tracking performance of these unobservable states. Note that this choice of error metrics reflects the role of proprioceptive estimators properly, which is responsible for providing reliable velocity, orientation estimation, and local position estimation to enable stable control of the robot. Since we collected five different experimental data, all metrics used in this section implicitly stand for the average over these five experimental data and its corresponding standard deviation. From these metrics and experimental data, the estimation performance is mainly investigated from two viewpoints: the merit of the invariant framework compared with the noninvariant framework in Section VII-A and the benefit of smoothing framework in invariant frameworks in Section VII-B. We then review the offline estimation performance of the estimators in Section VII-C, and finally, in VII-D, estimation performances in the environments with static contact events are presented for the ablation study.

A. Effect of the Group-Affine Property in Smoothing Framework

In this section, we test the advantage of the group-affine property for smoothing algorithms. Smoothing algorithms model
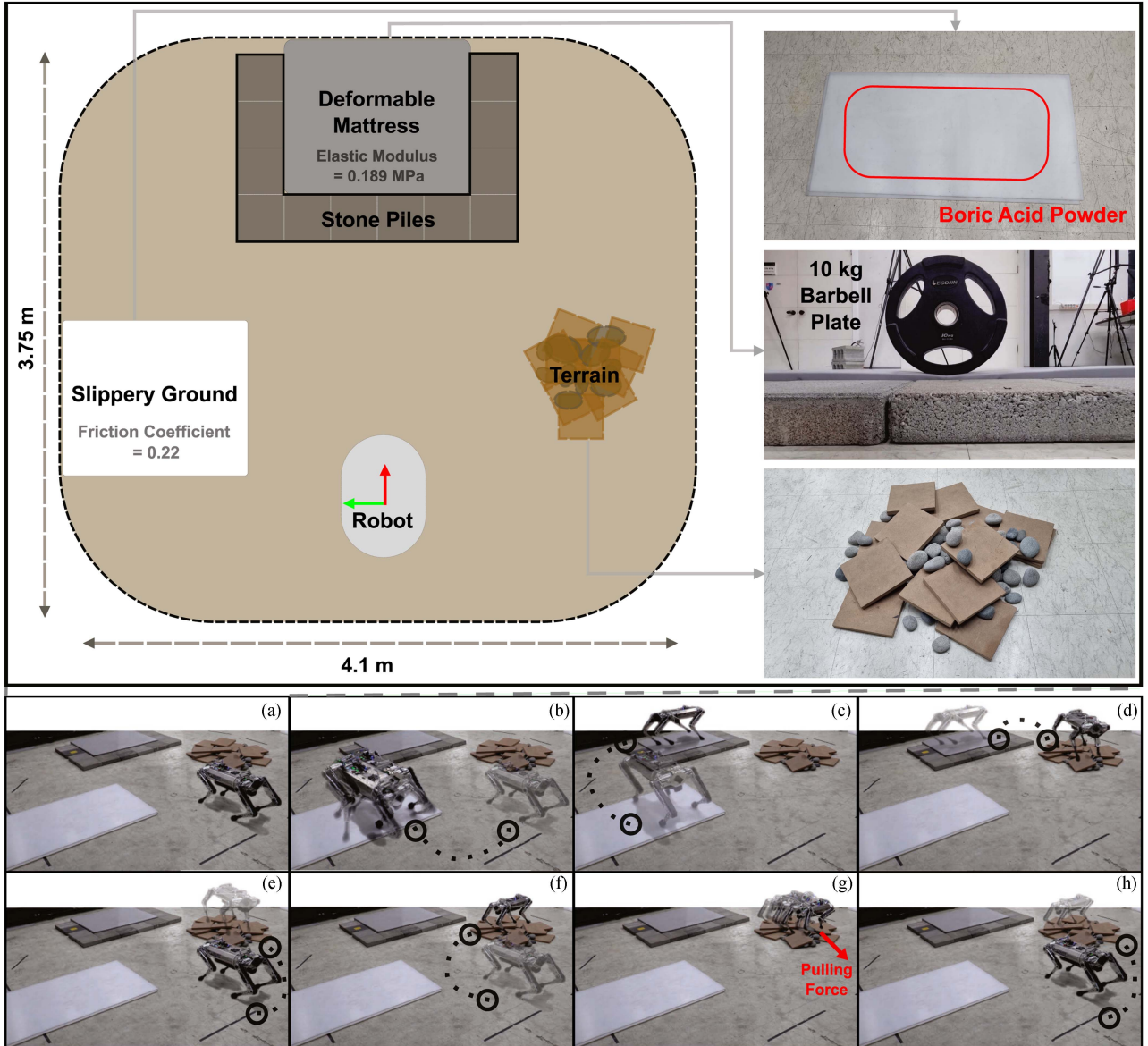
Fig. 7. Sequential illustration of the experiment procedure, where the procedure sequence is left to right, top to bottom, denoted as the alphabet order. (a) Robot starts at the initial position. (b) Robot moves to the slippery board and moves around the area to encounter slipping situations. (c) Robot moves to the deformable mattress environment and walks on it to face the z-directional movement of the ground. (d) Robot moves to the unstable terrain and passes the area. (e) Robot comes back to the initial position. (f) Robot moves to the unsteady terrain again. (g) Strong pulling forces are applied several times to the robot to check the robustness of the estimator. (h) Robot comes back to the initial position. Overall experiment time is about two minutes, and the experiment is held five times for statistical credibility.

probability distribution as nonlinear cost functions and resolve it by leveraging nonlinear optimization. Consequently, the linearization of the nonlinearity provokes the threat of falling into a suboptimum or even divergence of the solution. In our algorithm, the observation cost function is one of the factors that could induce nonlinearity in the optimization. Reviewing the state definition of PIS and NIS, they cannot formulate the group-affine observation (28) since they handle only a part of their states on a manifold. As a result, the Jacobian of the residual function about the orientation perturbation ϕ becomes state-dependent for both PIS and NIS formulation. In the case of PIS, we have already seen in (79) that the Jacobian about ϕ is state-dependent. For NIS, the residual and the Jacobian of the residual about ϕ are as

follows, which are dependent on the current estimated state:

$$\mathbf{r}_{\text{Obs}} = (\bar{\mathbf{R}}_{i+1} fk(\tilde{\mathbf{q}}_{i+1}) + \bar{\mathbf{p}}_{i+1}) - (\bar{\mathbf{R}}_i fk(\tilde{\mathbf{q}}_i) + \bar{\mathbf{p}}_i)$$

$$\mathbf{J}_{\phi_{i+1}} = -(\bar{\mathbf{R}}_{i+1} fk(\tilde{\mathbf{q}}_{i+1}))^\wedge, \quad \mathbf{J}_{\phi_i} = (\bar{\mathbf{R}}_i fk(\tilde{\mathbf{q}}_i))^\wedge. \quad (81)$$

However, in the case of IS, manifold state definition, which satisfies invariant observation form (28) results in the state-independent Jacobian about ϕ for the observation residual function (70). The state-dependent Jacobians of noninvariant formulations may cause an estimation result to fall in a suboptimal point, resulting in distortion of estimated states when the observation cost model is linearized at the operating point far from the ground truth. Note that the legged robot state estimation is
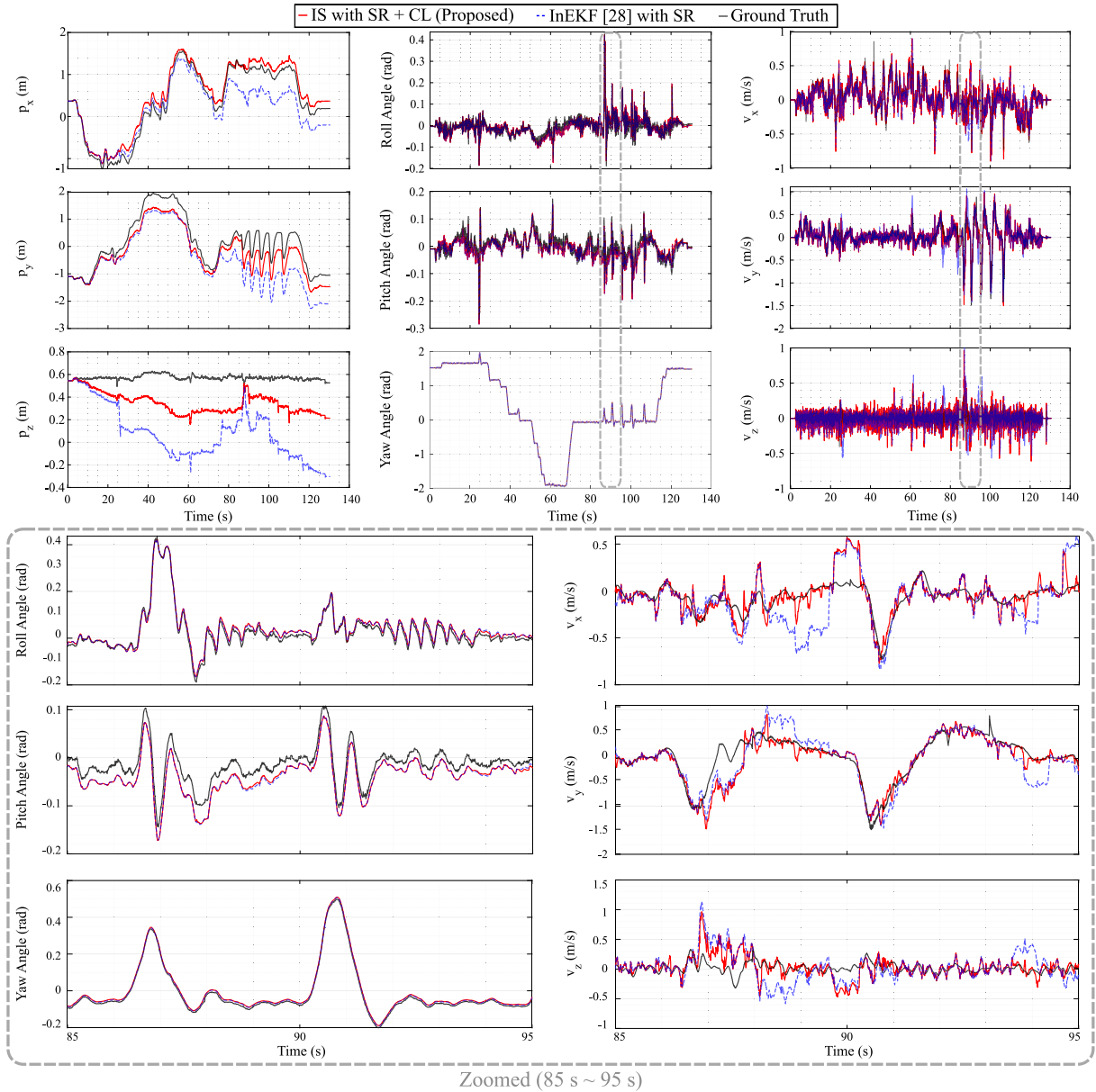
Fig. 8. Online estimation result comparison of IS and InEKF for the experiment described in Fig. 7 along with the ground truth. Figures depict the x, y, and z-directional position, orientation, and body frame velocity, respectively. During the experiment, a strong pulling force is applied multiple times. The grey-dotted box indicates the estimation results of observable states under the first two disturbances during 85–95 s and magnified in the bottom part of the figure.

affected by the observation cost model combined with the static foot contact assumption, and one caveat for the system is that the static foot contact assumption is often violated. When a robot slips, the static foot contact assumption can set the operating point far from the true state. We plotted the orientation estimation results of the smoothers in Fig. 10, especially when the robot is slipping in the situation described in Fig. 7(g). To exclusively test the effect of different state definitions, SR and CL strategies are turned OFF for all algorithms. In Fig. 10, we can see that the estimators equipped with noninvariant observation deviate larger than IS from the ground truth as the robot slips. As a result, IS produced a lower average level of orientation ATE in the whole five experimental data than the others.

To elaborate on the effect of the state-dependent Jacobian, we compare not only the estimation results but also the optimization

processes of the estimators in Fig. 9. For a fair comparison, we run the estimators as a full batch without backtracking. The estimators are run during the same time span in Fig. 10 since the estimators frequently confront the violation of the static foot contact assumption in this period. Furthermore, the convergence condition and the maximum iteration number are modified to 10^{-5} and 100 times to guarantee full convergence. The estimators are initialized with the ground truth at the start of the given time span, whereas the prior covariance is set as the diagonal components of the covariance matrix (inverse of the marginalized Hessian) of the fixed-lag IS at that time to reproduce the similar prior distribution to Fig. 10. To examine the effect of the state-independency of the Jacobians during optimization processes, we plotted the cost ratio and iteration number that compares the final cost after the convergence of
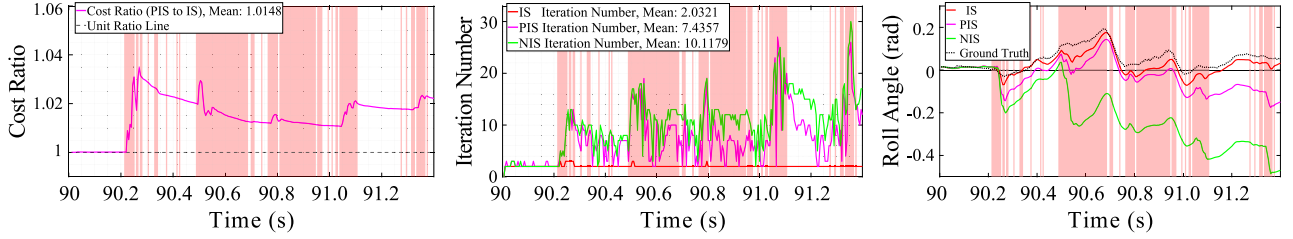
Fig. 9.   Comparison of the cost ratio, iteration numbers, and roll estimation results of IS, PIS, and NIS when the robot slips. The red-shaded area represents the time span of the ground truth slip when the velocity of the foot exceeds 0.5 m/s. The time period in the graphs is the same as that in Fig. 10. The estimators are run in a full batch setting without backtracking and are warranted to converge. The left, middle, and right figures are the cost, iteration number, and roll estimation, respectively. In the case of the cost, only PIS and IS are compared due to the difference in cost definitions between them and NIS. In the legend of the figures, except for the estimation result plot, the mean value of the measure over the given period is described.
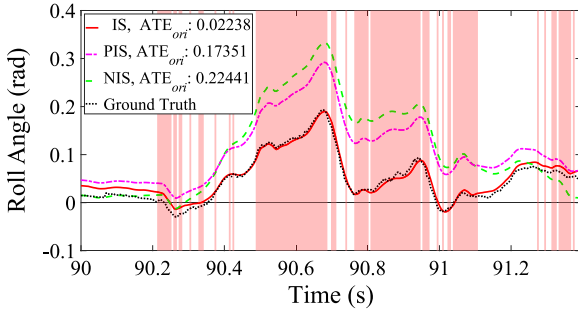

Fig. 10.   Zoomed-in estimation result of the roll angle of IS, PIS, and NIS when the robot slips. The red-shaded area represents the time span of the ground truth slip when the velocity of the foot exceeds 0.5 m/s. In order to exclusively test the effect of different state definitions, SR and CL strategies are turned OFFfor all algorithms. Also, each estimator's orientation ATE over five experimental data is written in the legend.

each estimator and the iteration required for convergence. In the case of the cost ratio, only PIS and IS are compared since they have similar cost definitions, as we have seen in Section VI. In Fig. 9, the left figure shows the ratio of the cost of PIS to IS after the convergence at each timestep. Note that the cost of PIS and IS after convergence has been almost the same until the first slip at the time near 90.2 s since their cost definitions are comparable. Nevertheless, the ratio increases when the static contact foot assumption is damaged, marked by the red region. We can also see that the iteration number of PIS in the middle graph increases up to 25 times larger than that of IS when the robot slips. Consistently, in the right plot for the roll orientation estimation results, the estimation result of PIS deviates more than IS when the robot slips. These results allow us to infer that the state-dependent Jacobians of PIS damage the optimization process and make a larger error. Although we can not directly compare the cost of NIS with that of IS due to the different cost definitions, we can identify that NIS also shows an immense number of iterations and a much larger estimation error in the figures.

B. Advantages of Smoothing in Invariant Frameworks

In this section, we will examine the merit of the smoothing framework introduced by the SR and CL strategies for invariant frameworks. Since invariant frameworks' Jacobian do not depend on the current linearization point, the smoothing approach has less chance to wield its advantage over filtering in terms

TABLE III
COMPARISON OF ATE AND 0.6 s RE (MEAN, THE EXCESS OVER THE MEAN VALUE OF THE PROPOSED ESTIMATOR, AND STANDARD DEVIATION) BETWEEN ESTIMATORS WITH VARIOUS STRATEGIES

	ATE_{ori}	ATE_{vel}	RE_{ori}	RE_{vel}	RE_{pos}
IS with SR + CL	**0.01984**	**0.19436**	**0.00855**	**0.22716**	**0.05241**
	(0.00266)	(0.01868)	(0.00423)	(0.12812)	(0.04362)
IS with SR	0.01989	0.21320	0.00857	0.23773	0.06251
	(+0.3%)	(+9.7%)	(+0.2%)	(+4.7%)	(+19.3%)
	(0.00264)	(0.02657)	(0.00426)	(0.13572)	(0.05269)
NIS with SR [18] + CL	0.02992	0.26461	0.00964	0.29358	0.07005
	(+50.8%)	(+36.1%)	(+12.7%)	(+29.2%)	(+33.7%)
	(0.00968)	(0.04209)	(0.00480)	(0.14133)	(0.05592)
NIS with SR [18]	0.02003	0.21365	0.00857	0.23773	0.06176
	(+1.0%)	(+9.9%)	(+0.2%)	(+4.7%)	(+17.8%)
	(0.00461)	(0.02683)	(0.00425)	(0.13513)	(0.05189)
InEKF [32] with SR	0.01989	0.23341	0.00859	0.24815	0.07260
	(+0.3%)	(+20.1%)	(+0.5%)	(+9.2%)	(+38.5%)
	(0.00267)	(0.02249)	(0.00426)	(0.14348)	(0.06096)
InEKF [32]	0.02245	0.21584	0.00869	0.27477	0.05862
	(+13.2%)	(+11.1%)	(+1.6%)	(+21.0%)	(+11.8%)
	(0.00109)	(0.01890)	(0.00436)	(0.16578)	(0.04195)

The Smallest mean error values are marked in bold.

of relinearization. However, when we leverage the SR and CL methods, the smoothing methodology has a chance to bring additional benefits by re-evaluating the past sliding information and including more CLs as the window extends. To demonstrate this merit, we investigated the estimation error of IS in Fig. 11 by increasing the window size from 1 to 15, whereas the InEKF inherently retains the window size of 1. The SR method is applied to both, but the CL method is applied only to the smoother since the past states are marginalized in the InEKF. The estimation error of the IS gradually decreases as the window size increases compared with the filter, where the velocity and the positional error decrease notably. Another aspect of the result becomes more noticeable when the static contact assumption does not hold. In Fig. 12, part of the body frame velocity error result is shown. The estimators face increasing estimation errors during the slippage of the feet. Nevertheless, the larger the window of the estimator is, the faster the error decreases. This is because IS with a larger window could detect SR status more accurately by re-estimating the previous trajectory in its window. The effect
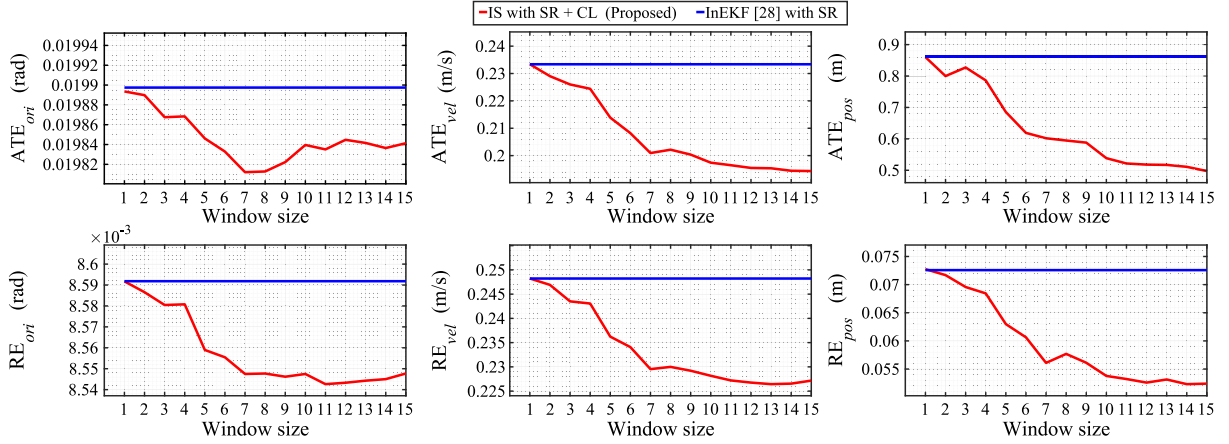
Fig. 11. Comparison between the estimation results of smoothing and filtering for invariant frameworks, varying the window size from 1 to 15. The top line illustrates ATE, and the bottom line depicts RE. Generally, the errors of IS decrease in the entire error plots as the window size becomes larger, thanks to the SR and CL methods. Still, the errors sometimes can increase even when the window size increases since the methods rely on the estimation results and, hence, can be affected by the erroneous estimation results in the past.
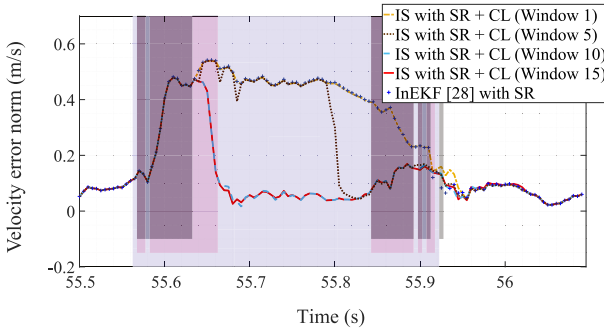

Fig. 12. Part of the body frame velocity error results from the estimators with different windows. The estimation results of InEKF are plotted by circle dotted lines. The grey, red, and blue-shaded areas represent the time span of ground truth slip, slip estimated by IS with SR + CL (Window 15), and slip estimated by InEKF, respectively. Each shaded region is illustrated with different y-axis values to ease the distinguishment.
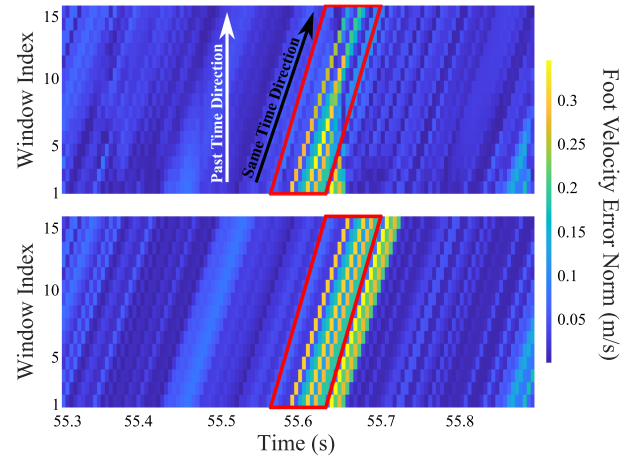

Fig. 13. Foot velocity error along the accumulating time window. The foot velocity errors within the window of the estimator at a certain timestep are "screenshotted" and listed along the y-axis. Therefore, the foot velocity error of a fixed timestep is tracked diagonally, from the bottom left to the top right. The red parallelogram highlights the slip events of the experiment. The top figure is for the proposed SR method, which re-evaluates the foot velocity. In contrast, the bottom figure is for the SR method in [18], which does not re-evaluate the foot velocity. Note that the brighter color on the figures represents a higher error against the true foot velocity calculated from the motion capture device and the encoder.

of re-evaluation can be seen more straightforwardly through Fig. 13, where the proposed SR method, which re-evaluates the foot velocity, can utilize richer information than the previous method, which does not re-evaluate. Also, in Fig. 12, we could identify that the grey-shaded area (true slip) is more matched with the red-shaded region (slip estimated by IS with window 15) than the blue-shaded area (slip estimated by InEKF). Another interesting result is presented in Fig. 14, which shows that the estimator output becomes similar to the straight integration of the IMU signal when all the feet are slipping. This result shows that the estimator mainly utilizes IMU measurements when the kinematic measurements are undesirable. In addition, we attest to the profit of the CL in Table III, by comparing smoothers equipped with and without CL. The window size of smoothers is fixed as 15. We can see in this table that IS with SR and CL shows better performance than IS equipped only with SR, especially in RE_{pos}. On the contrary, in the case of NIS, the CL method affects the estimation results adversely. We attribute this negative effect to the state-dependence of NIS, which could distort the estimation result when the robot is sliding with a

strong contact belief, as we have seen in the previous section. The CL method assumes lower covariance when the calculated foot speed and acceleration are lower than thresholds. It means that there could be erroneous CLs that presume lower contact uncertainty for a sliding foot. In such conditions, NIS is likely to produce estimation results with large errors.

C. Off-Line Comparison to the Existing Algorithms

In this section, we testify the profit of the suggestion in this article by comparing the estimation performance of the proposed algorithm, the existing invariant filter, and the NIS in Table III. The proposed algorithm in the top row shows the lowest error

TABLE IV
COMPARISON OF ATE AND 0.6 S RE (MEAN, THE EXCESS OVER THE MEAN VALUE OF THE PROPOSED ESTIMATOR, AND STANDARD DEVIATION) BETWEEN IS, INEKF, AND NIS WITH LEGGED ROBOT-SPECIFIC METHODS

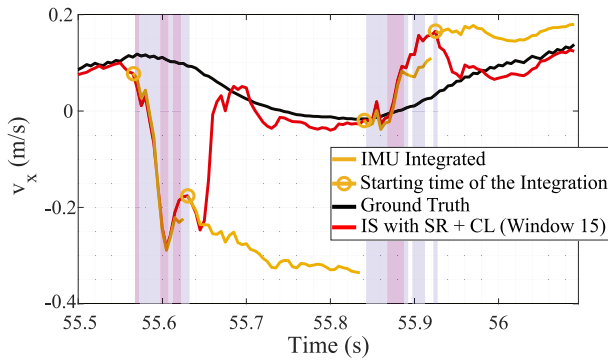|	Static Contact [Fig. 7(a), early (b)]					Dynamic Contact [Fig.7(g)]				
	ATE_{ori}	ATE	RE_{ori}	RE_{vel}	RE_{pos}	ATE_{ori}	ATE_{vel}	RE_{ori}	RE_{vel}	RE_{pos}
IS with SR + CL	0.01194	**0.10323**	0.00688	**0.13848**	**0.02734**	**0.02319**	**0.28826**	**0.01075**	**0.31882**	**0.08862**
	(0.00335)	(0.01307)	(0.00474)	(0.08226)	(0.02213)	(0.00403)	(0.02784)	(0.00523)	(0.17411)	(0.07386)
NIS with SR [18] + CL	0.01260	0.16010	0.00813	0.20779	0.03677	0.03616	0.36595	0.01207	0.38505	0.14999
	(+5.5%)	(+55.1%)	(+18.2%)	(+50.1%)	(+34.5%)	(+55.9%)	(+61.6%)	(+12.3%)	(+20.8%)	(+69.3%)
	(0.00307)	(0.01516)	(0.00570)	(0.12220)	(0.02934)	(0.01309)	(0.06012)	(0.00592)	(0.17519)	(0.12632)
InEKF [32] with SR	**0.01188**	0.10803	**0.00686**	0.14438	0.03443	0.02334	0.34129	0.01084	0.34535	0.11951
	(-0.5%)	(+4.6%)	(-0.3%)	(+4.3%)	(+25.9%)	(+0.6%)	(+18.4%)	(+0.8%)	(+8.3%)	(+34.9%)
	(0.00335)	(0.01446)	(0.00473)	(0.08566)	(0.02934)	(0.00392)	(0.01368)	(0.00527)	(0.18620)	(0.09818)

The smallest mean error values are marked in bold.

Fig. 14. Body frame x-directional velocity plots of the ground truth, IMU integrations, and the estimation results of the proposed IS. The four yellow IMU integration plots are produced by choosing four starting times and then integrating the IMU measurements from the estimated state values at each starting time. The blue region denotes the ground truth slip period with more than one leg slipping, whereas the red region marks the ground truth slip period with all feet slipping.

Fig. 15. Estimation result in the 160 m-long outdoor experiments. The robot walks along an ∞-shaped path with the same starting and end point at the yellow-circled point. The proposed algorithm estimates the trajectory of the robot, which is overlapped on the bird-eye view map as a red line.

for every metric, especially in velocity and position estimation. We can identify that the legged robot-specific methods are best exploited in the proposed algorithm, which captures the advantage of both the group-affine property and the smoother framework at once.

D. Comparison Against Contact Conditions and Estimation in Outdoor Experiment

In this last section, we test the profit of the proposed IS while comparing against contact conditions and show the result for the outdoor experiment. To identify the degree of the benefit claimed in this article in different environments, we perform the error analysis to the trajectory segments extracted from the indoor experiment described in Fig. 7. In Table IV, the performance of the estimators in different contact conditions is displayed. The estimation performance of trajectory where static contact event occurs mostly, which corresponds to the experimental procedure Fig. 7(a) and the beginning part of (b), is in the left side of the table and that for trajectory where dynamic contact event occurs mostly, which is featured by Fig. 7(g), is in the right side of the table. Unsurprisingly, the error is much smaller
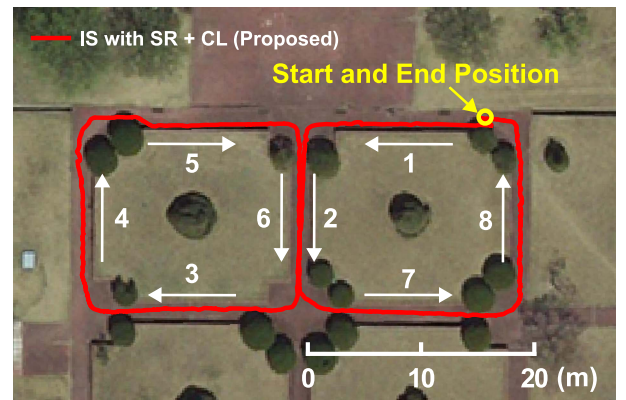
under the static contact condition than under the dynamic contact condition. A noteworthy point in this table is that the differences between the proposed IS and the other algorithms become larger under the dynamic contact condition. This result agrees with the nature of the benefit of invariant smoother that we have covered; invariant smoother is more robust than its noninvariant counterpart when it handles dynamic contact events, and it can leverage the SR method better than its filtering counterpart. Since both advantages appear when the robot is under dynamic contact, it is understandable that the proposed algorithm shows a larger difference from the others, especially when the robot is slipping.

In the outdoor experiment, the robot is commanded to make a 160 m-long ∞-shaped loop trajectory and return to where it started. The robot is controlled by the online feedback of the proposed invariant smoother. The estimation performance of our proposed estimator in the experiment is drawn in Fig. 15, where the estimated trajectory of the proposed estimator is well overlapped with the bird-eye view map. Since the ground truth data of this experiment is unavailable, we do not perform a detailed analysis on velocity, but it is worth noting that the proposed estimator can give a precise position estimation result when there are no dynamic contact events.

VIII. CONCLUSION

In this article, a new algorithm for legged robot state estimation was proposed by combining the group-affine property and smoothing framework. The proposed algorithm was implemented in the quadruped robot *KAIST HOUND*, and using the online estimation of the proposed algorithm, the robot was controlled to overcome the environment with unstable contacts. Further exploiting the data collected from the experiments, we have discovered that the state-independent Jacobian of the group-affine property is beneficial in the nonlinear optimization process by reducing the required iteration number. As a result, IS showed lower orientation estimation error than PIS and NIS. Meanwhile, we leveraged the static contact foot assumption of the legged robot state estimator by introducing SR and CL methodologies, which also take advantage of the smoother frameworks. Thanks to the advantages induced by the group-affine property and smoothing frameworks, our proposed IS recorded the lowest error among the baseline legged robot state estimators proposed in [32] and [18] for the collected data.

Future works may include adding exteroceptive sensor measurements, visual loop-closure, and nonlinear constraints [57] to enjoy the further benefit of the smoothing framework with the group-affine property. In addition, the frequency of the estimator can be further increased by leveraging the state-independent sparsity characteristic of the Hessian matrix, where the frequency was restricted to 200 Hz due to the limit of our robot's computation resources in this work. On the other hand, the uncertainty modeling of the contact dynamics for legged robots can be described precisely by introducing terrain information or the neural network-based representation. Finally, the overall derivation of the propagation cost model can be calculated using the discrete-time domain dynamics equations, which can be mathematically more thorough than the discretization from the continuous dynamics.

REFERENCES

[1] M. Bloesch et al., "State estimation for legged robots: Consistent fusion of leg kinematics and IMU," in *Proc. Robot. Sci. Syst. Conf.*, 2012, pp. 17–24.

[2] S. Hong, J.-H. Kim, and H.-W. Park, "Real-time constrained nonlinear model predictive control on SO(3) for dynamic legged locomotion," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2020, pp. 972–979.

[3] Y.-H. Shin et al., "Design of KAIST HOUND, a quadruped robot platform for fast and efficient locomotion with mixed-integer nonlinear optimization of a gear train," in *Proc. Int. Conf. Robot. Autom.*, 2022, pp. 6614–6620.

[4] S. Thrun, "Probabilistic robotics," *Commun. ACM*, vol. 45, no. 3, pp. 52–57, 2002.

[5] F. Dellaert, "Factor graphs and GTSAM: A hands-on introduction," Georgia Inst. Technol., Atlanta, GA, USA, Tech. Rep. GT-RIM-CP&R-2012-002, 2012.

[6] T. D. Barfoot, *State Estimation for Robotics*. Cambridge, U.K.: Cambridge Univ. Press, 2017.

[7] F. Dellaert et al., "Factor graphs for robot perception," *Found. Trends Robot.*, vol. 6, no. 1–2, pp. 1–139, 2017.

[8] F. Dellaert and M. Kaess, "Square root SAM: Simultaneous localization and mapping via square root information smoothing," *Int. J. Robot. Res.*, vol. 25, no. 12, pp. 1181–1203, 2006.

[9] M. Kaess, H. Johannsson, R. Roberts, V. Ila, J. J. Leonard, and F. Dellaert, "iSAM2: Incremental smoothing and mapping using the Bayes tree," *Int. J. Robot. Res.*, vol. 31, no. 2, pp. 216–235, 2012.

[10] C. Forster, L. Carlone, F. Dellaert, and D. Scaramuzza, "On-manifold preintegration for real-time visual–inertial odometry," *IEEE Trans. Robot.*, vol. 33, no. 1, pp. 1–21, Feb. 2017.

[11] 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 *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2018, pp. 3783–3790.

[12] R. Hartley et al., "Legged robot state-estimation through combined forward kinematic and preintegrated contact factors," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2018, pp. 4422–4429.

[13] G. P. Huang, A. I. Mourikis, and S. I. Roumeliotis, "An observability-constrained sliding window filter for SLAM," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2011, pp. 65–72.

[14] T. Qin, P. Li, and S. Shen, "VINS-Mono: A robust and versatile monocular visual-inertial state estimator," *IEEE Trans. Robot.*, vol. 34, no. 4, pp. 1004–1020, Aug. 2018.

[15] L. Von Stumberg, V. Usenko, and D. Cremers, "Direct sparse visual-inertial odometry using dynamic marginalization," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2018, pp. 2510–2517.

[16] D. Wisth, M. Camurri, and M. Fallon, "Robust legged robot state estimation using factor graph optimization," *IEEE Robot. Autom. Lett.*, vol. 4, no. 4, pp. 4507–4514, Oct. 2019.

[17] D. Wisth, M. Camurri, and M. Fallon, "Preintegrated velocity bias estimation to overcome contact nonlinearities in legged robot odometry," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2020, pp. 392–398.

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

[19] 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 Robot. Autom. Lett.*, vol. 7, no. 2, pp. 4456–4463, Apr. 2022.

[20] M. Fallon, M. Antone, N. Roy, and S. Teller, "Drift-free humanoid state estimation fusing kinematic, inertial and LiDAR sensing," in *Proc. IEEE/RAS Int. Conf. Humanoid Robots*, 2014, pp. 112–119.

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

[22] 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," *Int. J. Robot. Res.*, vol. 35, no. 6, pp. 631–653, 2016.

[23] M. Camurri, M. Ramezani, S. Nobili, and M. Fallon, "Pronto: A multi-sensor state estimator for legged robots in real-world scenarios," *Front. Robot. AI*, vol. 7, 2020, Art. no. 68.

[24] S. Teng, M. W. Mueller, and K. Sreenath, "Legged robot state estimation in slippery environments using invariant extended Kalman filter with velocity update," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2021, pp. 3104–3110.

[25] H. Strasdat, J. Montiel, and A. J. Davison, "Scale drift-aware large scale monocular slam," *Robot., Sci. Syst. VI*, vol. 2, no. 3, pp. 73–80, 2010.

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

[27] D. Wisth, M. Camurri, and M. Fallon, "VILENs: Visual, inertial, lidar, and leg odometry for all-terrain legged robots," *IEEE Trans. Robot.*, vol. 39, no. 1, pp. 1–18, Feb. 2023.

[28] 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 *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2013, pp. 6058–6064.

[29] F. Jenelten, J. Hwangbo, F. Tresoldi, C. D. Bellicoso, and M. Hutter, "Dynamic locomotion on slippery ground," *IEEE Robot. Autom. Lett.*, vol. 4, no. 4, pp. 4170–4176, Oct. 2019.

[30] A. Barrau and S. Bonnabel, "The invariant extended Kalman filter as a stable observer," *IEEE Trans. Autom. Control*, vol. 62, no. 4, pp. 1797–1812, Apr. 2017.

[31] A. Barrau and S. Bonnabel, "Invariant Kalman filtering," *Annu. Rev. Control, Robot., Auton. Syst.*, vol. 1, no. 1, pp. 237–257, 2018.

[32] R. Hartley, M. Ghaffari, R. M. Eustice, and J. W. Grizzle, "Contact-aided invariant extended Kalman filtering for robot state estimation," *Int. J. Robot. Res.*, vol. 39, no. 4, pp. 402–430, 2020.

[33] P. Chauchat, A. Barrau, and S. Bonnabel, "Invariant smoothing on Lie groups," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2018, pp. 1703–1710.

[34] A. Walsh, J. Arsenault, and J. R. Forbes, "Invariant sliding window filtering for attitude and bias estimation," in *Proc. Amer. Control Conf.*, 2019, pp. 3161–3166.

[35] J. Huai, Y. Lin, Y. Zhuang, and M. Shi, "Consistent right-invariant fixed-lag smoother with application to visual inertial SLAM," in *Proc. AAAI Conf. Artif. Intell.*, 2021, pp. 6084–6092.

[36] B. Katz, J. D. Carlo, and S. Kim, "Mini cheetah: A platform for pushing the limits of dynamic quadruped control," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2019, pp. 6295–6301.

[37] J. Hwangbo, C. D. Bellicoso, P. Fankhauser, and M. Hutter, "Probabilistic foot contact estimation by fusing information from dynamics and differential/forward kinematics," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2016, pp. 3872–3878.

[38] M. Camurri et al., "Probabilistic contact estimation and impact detection for state estimation of quadruped robots," *IEEE Robot. Autom. Lett.*, vol. 2, no. 2, pp. 1023–1030, Apr. 2017.

[39] T.-Y. Lin, R. Zhang, J. Yu, and M. Ghaffari, "Legged robot state estimation using invariant Kalman filtering and learned contact events," in *Proc. 5th Annu. Conf. Robot Learn.*, 2021, pp. 1057–1066.

[40] P. Varin, "Estimation and planning for dynamic robot behaviors," Harvard University, 2021.

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

[42] R. Buchanan, M. Camurri, F. Dellaert, and M. Fallon, "Learning inertial odometry for dynamic legged robot state estimation," in *Proc. 5th Annu. Conf. Robot Learn.*, 2021, pp. 1575–1584.

[43] G. S. Chirikjian, "Stochastic models, information theory, and lie groups," *Classical Results and Geometric Methods*, vol. 1, Berlin, Germany: Springer, 2009.

[44] G. S. Chirikjian, "Stochastic models, information theory, and Lie groups," *Analytic Methods and Modern Appl.*, vol. 2, Berlin, Germany: Springer, 2011.

[45] E. Eade, "Lie groups for 2D and 3D transformations," 2013. [Online]. Available: http://ethaneade.com/lie.pdf

[46] J. Sola, J. Deray, and D. Atchuthan, "A micro Lie theory for state estimation in robotics," 2018, arXiv:1812.01537.

[47] R. Smith, M. Self, and P. Cheeseman, "A stochastic map for uncertain spatial relationships," in *Proc. 4th Int. Symp. Robot. Res.*, 1988, pp. 467–474.

[48] J. Castellanos, J. Montiel, J. Neira, and J. Tardos, "The SPmap: A probabilistic framework for simultaneous localization and map building," *IEEE Trans. Robot. Autom.*, vol. 15, no. 5, pp. 948–952, Oct. 1999.

[49] M. Dissanayake, P. Newman, S. Clark, H. D.-Whyte, and M. Csorba, "A solution to the simultaneous localization and map building (SLAM) problem," *IEEE Trans. Robot. Automat.*, vol. 17, no. 3, pp. 229–241, Jun. 2001.

[50] S. T. Smith, "Optimization techniques on Riemannian manifolds," *Fields Inst. Commun.*, vol. 3, no. 3, pp. 113–135, 1994.

[51] P.-A. Absil, C. G. Baker, and K. A. Gallivan, "Trust-region methods on Riemannian manifolds," *Found. Comput. Math.*, vol. 7, no. 3, pp. 303–330, 2007.

[52] L. Armijo, "Minimization of functions having Lipschitz continuous first partial derivatives," *Pacific J. Math.*, vol. 16, no. 1, pp. 1–3, 1966.

[53] G. Wu and K. Sreenath, "Variation-based linearization of nonlinear systems evolving on $SO(3)$ and $\mathbb{S}^2$," *IEEE Access*, vol. 3, pp. 1592–1604, 2015.

[54] M. Chignoli and P. M. Wensing, "Variational-based optimal control of underactuated balancing for dynamic quadrupeds," *IEEE Access*, vol. 8, pp. 49 785–49 797, 2020.

[55] S. Leutenegger, S. Lynen, M. Bosse, R. Siegwart, and P. Furgale, "Keyframe-based visual–inertial odometry using nonlinear optimization," *Int. J. Robot. Res.*, vol. 34, no. 3, pp. 314–334, 2015.

[56] Z. Zhang and D. Scaramuzza, "A tutorial on quantitative trajectory evaluation for visual(-inertial) odometry," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2018, pp. 7244–7251.

[57] C. Rao, J. Rawlings, and D. Mayne, "Constrained state estimation for nonlinear discrete-time systems: Stability and moving horizon approximations," *IEEE Trans. Autom. Control*, vol. 48, no. 2, pp. 246–258, Feb. 2003.

Ziwon Yoon received the bachelor's degree in mechanical engineering in Korea Advanced Institute of Science and Technology, Daejeon, Republic of Korea, in 2022.

He was a Research Engineer with the Dynamic Robot Control and Design Lab (DRCD Lab), Korea Advanced Institute of Science and Technology, under the supervision of Prof. Hae-Won Park. His research interests include state estimation, SLAM, legged robots, optimal control, and reinforcement learning.

Joon-Ha Kim (Graduate Student Member, IEEE) received the bachelor's degree in mechanical engineering from Hanyang University, Seoul, Republic of Korea, in 2017, and the M.Sc. degree in humanoid navigation from the Department of the Mechanical Engineering, Korea Advanced Institute of Science and Technology, Daejeon, Republic of Korea, in 2019, where he is currently working toward the Ph.D. degree under the supervision of Prof. Hae-Won Park.

His research interests include state estimation, legged robot navigation, optimal control, and reinforcement learning.

Hae-Won Park (Member IEEE) received the B.S. and M.S. degrees from Yonsei University, Seoul, South Korea, in 2005 and 2007, respectively, and the Ph.D. degree from the University of Michigan, Ann Arbor, MI, USA, in 2012, all in mechanical engineering.

He is currently an Associate Professor of mechanical engineering with the Korea Advanced Institute of Science and Technology, Daejeon, South Korea. His research interests include the intersection of control, dynamics, and mechanical design of robotic systems, with special emphasis on legged locomotion robots.

Dr. Park was the recipient of the 2018 National Science Foundation (NSF) CAREER Award and NSF most prestigious awards in support of early-career faculty.