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

Online Determination of Legged Kinematics

Chinmay Burgul, Woosik Lee, Patrick Geneva, and Guoquan Huang

***Abstract*— Legged robots are emerging, and legged locomotion is in critical need, which requires precise leg-body kinematics to execute control commands or plan motion trajectories. This paper proposes online state estimation to determine legged kinematics of robots with an arbitrary number of legs, which includes the kinematic parameters of the leg-body transformation, time offset and the leg link lengths. In particular, we advocate an in-place dance gait for kinematic determination where the toes remain static on the ground and serve as static landmarks as in SLAM. As a visual-inertial sensor is typically available onboard robot and located at the floating base, we leverage efficient MSCKF-based visual-inertial navigation to estimate legged kinematics. To this end, we analytically derive the legged kinematic measurements and tightly fuse them along with visual-inertial measurements for MSCKF update of both the leg's kinematics and body's motion. The proposed method has been extensively validated in both simulations and experiments with different quadrupeds, showing its robustness and accuracy.**

I. Introduction

Due to their stability and traversability in complex terrains, legged robots have demonstrated potential in various applications such as last-mile delivery and search and rescue [1]. Precise knowledge of the legged kinematics is *fundamental* to unlock the agile and robust locomotion capabilities of these high-DOF under-actuated legged robots. In particular, model-based locomotion controllers that presumably are most widely used in real robots today require kinematic parameters to plan body/foot trajectory and execute actuation commands [2]. On the other hand, a state estimator which is a prerequisite for the locomotion controller can benefit from fusing the legged kinematic constraints, if they are accurate, otherwise, detrimental effects are expected [3].

Determining legged kinematics typically involves either costly precision manufacturing along with tolerance validation to ensure the CAD model serves as ground truth or time-consuming offline manual calibration and validation processes. When using it, one may simply rely on the manufacturer-provided 3D model or manual measurements (if possible), which however can be imprecise or obsoleted due to manufacturing/operating variations or human errors. These unaccounted errors can propagate through the kinematic chain and significantly degrade the estimator/controller performance, possibly failing the robot's mission.

To address these issues, this work aims to design an easy and user-friendly procedure and method that can accurately and reliably determine the kinematics of any legged robot in an online real-time fashion. In particular, a legged robot performs in-place dance motions to identify the kinematic parameters, including the spatial and temporal transformation between the legs and body frames as well as the leg link lengths, wherein its floating base excites all 6DOF motions. Note that such fully-excited dance motion is essential to ensure the observability of the kinematic parameters, in analogy to the IMU-camera calibration in visual-inertial navigation [4]. We find that a 25-second in-place dance of the legged robot can be adequate to achieve rapid convergence of the kinematic parameters, which is extremely easy to conduct without any requirement on space or terrains. Note that this in-place motion effectively eliminates the uncertainty associated with foot contact, slippage, and accounting for dynamic friction in contact modeling [5], which is inevitable during walking or jumping but notoriously hard to address properly. With this motion, we fully utilize the measurements of the visual-inertial sensor (which is mounted on the robot base and assumed to coincide with the body frame) to build an efficient and consistent multi-state constraint Kalman filter (MSCKF)-based estimator [6] to compute the kinematic parameters along with the states.

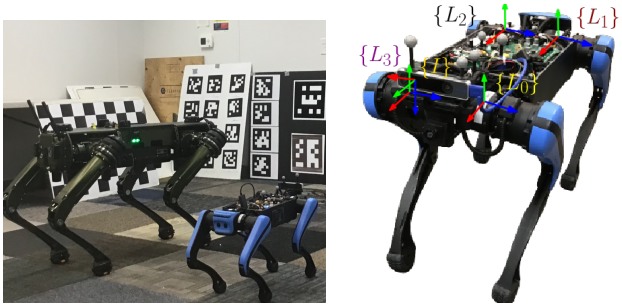

Fig. 1: Left: Our quadrupeds (Ghost Vision 60 and Jueying Lite 2) are dancing to determine their kinematic parameters. Right: The leg frames $\{L_i\}$ and body (IMU) frame $\{\mathrm{I}\}$.

The main contributions of this work are the following:

- We design an easy-to-use procedure with in-place dance motions to fast determine legged kinematics which is essential for legged locomotion (and state estimation). We are among the first to identify all the kinematic parameters including the spatio-temporal transformations (6DOF rigid transformation and time offset) between the body and all legs along with their link length.
- We develop a tightly-coupled efficient MSCKF-based state estimator to determine the kinematic parameters. In turn, our estimator models and fuses the leg kinematic constraints to improve state estimates and thus locomotion control. This proposed method is general and can be extended to any number of legs.
- We validate the proposed approach in Monte-Carlo simulations using our own legged simulator, and also on two different quadruped robots, showing accurate

This work was partially supported by the University of Delaware (UD) College of Engineering and NSF MRI-2018905.

The authors are with the Robot Perception and Navigation Group (RPNG), University of Delaware, Newark, DE 19716, USA. {cmburgul, woosik, pgeneva, ghuang}@udel.edu

kinematic determination and its repeatability, as well as that the found kinematics improves state estimation.

II. Related Work

Kinematic identification has been extensively studied in robotic manipulators, which typically uses costly external sensors in providing high-quality motion information [7]. Hand-eye calibration methods [8], [9] remove the need of external sensors and have been used to determine the kinematics of humanoid robots [10], [11]. This idea was also used for foot-eye calibration of quadruped robots, i.e., finding the extrinsic transformation between the camera and legs along with the leg encoder offsets and link lengths [12]. As the closest to our work, an offline batch estimation method was introduced to estimate the kinematic parameters including the transformation between the IMU and the body frame and the link lengths [13]. However, these kinematic parameters may not remain constant and may change over time due to changes in payload, sensor attachment changes, mechanical vibrations, and wear over time. As such, online kinematic calibration was studied recently in [14], [15], which also leverages visual-inertial navigation as we do in this work but focuses only on the one (last) leg link length along with observability analysis. In contrast, our proposed approach online determines the full kinematics including the spatial transformations and time offsets between all the legs and body frame and all the leg link parameters.

As visual-inertial estimation is a key enabler for the proposed method, we here also provide a brief overview of visual-inertial navigation systems (VINS), which have seen significant research efforts in the last decade [16]. There exists some work leveraging VINS for system identification. For example, in [17] dynamic parameters of an aerial vehicle were determined along with VINS estimation while wheel odometry kinematics of a ground vehicle was estimated based on VINS in [18]. However, challenges emerge when leveraging VINS on legged robots for kinematic determination. In particular, the impact shocks generated by footsteps can introduce motion blur in images and introduce additional noise into IMU measurements. Nevertheless, if the legged kinematics is accurate, fusing the kinematic constraints into VINS can improve estimation performance [3], [19]–[24]. This further motivates us to determine accurate legged kinematics and fuse its constraints in state estimation.

III. Legged Kinematics

In this section, we present a general kinematic model of a single leg, while it can be extended to a legged robot with an arbitrary number of legs, such as quadrupeds or hexapods.

Fig. 2 illustrates the standard kinematic chain and frame of reference of a robot leg. The leg frame $\{L\}$ is rigidly connected to the robot's body frame of reference $\{I\}$ where the high-level motion planner, low-level locomotion controller, or state estimator is typically operating, which clearly necessitates finding the leg kinematic relationships for the legged robot's estimation and control. The leg $\{L\}$ often aligns with the abduction joint $\{A\}$ whose orientation is altered by activating the first joint. As seen from Fig. 2, the three links connect the four joint frames of abduction $\{\mathrm{A}\}$, hip $\{\mathrm{H}\}$, knee $\{\mathrm{K}\}$, and toe $\{\mathrm{T}\}$. The top three joints are actuated with joint encoders to provide absolute joint angle measurements:

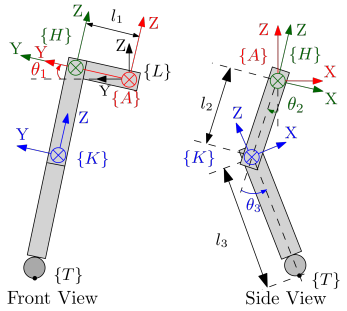

Fig. 2: Kinematics of a robot leg with three joint frames: abduction $\{\mathrm{A}\}$, hip $\{\mathrm{H}\}$, knee $\{\mathrm{K}\}$, and toe frame $\{\mathrm{T}\}$, and three links: hip (l_1), thigh (l_2) and shank (l_3).

$$\theta_{m,i} = \theta_i + n_{\theta_i}, \quad \forall i \in \{A, H, K\} \tag{1}$$

where θ_i is the true angle and n_{θ_i} is the white Gaussian noise. Stacking them yields the measurement vector $\boldsymbol{\theta}_m$.

Foot placement is critical for legged locomotion [2], [25], which requires the kinematic chain to determine the time-varying position of the foot (toe) $\{T\}$ with respect to the leg $\{L\}$ and thus the body $\{I\}$:

$$\begin{aligned} &{}^{I}\mathbf{p}_T = \\ &{}^{I}\mathbf{p}_L + {}^{L}_{I}\mathbf{R}^\top \left[{}^{L}\mathbf{p}_A + {}^{L}_{A}\mathbf{R}({}^{A}\mathbf{p}_H + {}^{A}_{H}\mathbf{R}({}^{H}\mathbf{p}_K + {}^{H}_{K}\mathbf{R}\,{}^{K}\mathbf{p}_T)) \right] \end{aligned} \tag{2}$$

where the 6DOF rigid transformation between the leg and body frames $\{{}^{I}\mathbf{p}_L, {}^{L}_{I}\mathbf{R}\}$ needs to be estimated, e.g., by leveraging visual-inertial systems, and the other transformations between the associated joint frames are given by:

$$ {}^{L}\mathbf{p}_A = \mathbf{0}_{3\times 1}, \qquad {}^{L}_{A}\mathbf{R} = \mathrm{Exp}(\theta_A \mathbf{k}_A) \tag{3}$$

$$ {}^{A}\mathbf{p}_H = l_1 \mathbf{k}_1 \ , \qquad {}^{A}_{H}\mathbf{R} = \mathrm{Exp}(\theta_H \mathbf{k}_H) \tag{4}$$

$$ {}^{H}\mathbf{p}_K = l_2 \mathbf{k}_2 \ , \qquad {}^{H}_{K}\mathbf{R} = \mathrm{Exp}(\theta_K \mathbf{k}_K) \tag{5}$$

$$ {}^{K}\mathbf{p}_T = l_3 \mathbf{k}_3 \tag{6}$$

In the above expressions, $\mathbf{k}_A$, $\mathbf{k}_H$, and $\mathbf{k}_K$ are the rotation axis of the joints. $\mathbf{k}_1$, $\mathbf{k}_2$, and $\mathbf{k}_3$ are the unit direction vector between the joints. $\mathrm{Exp}(\cdot)$ is the $SO(3)$ matrix exponential function to represent the rotation [26].

It becomes evident from the leg forward kinematic chain (2) that the rigid transformation $\{{}^{L}\mathbf{p}_I, {}^{L}_{I}\mathbf{R}\}$ is one of the key parameters required to calibrate accurately. Moreover, note that the leg is assumed to be aligned with the abduction actuator/encoder, which could be temporally (not only spatially) different from the body's IMU due to improper hardware triggering, transmission delays, or clock synchronization errors. This necessitates estimation of the timeline misalignment (time offset) between the body's IMU and the leg's encoders. We thus model this varying time offset ${}^{I}t_L$ as:

$$ {}^{I}t_k = {}^{L}t_k + {}^{L}t_I \tag{7}$$

where ${}^{I}t_k$ and ${}^{L}t_k$ are the timestamps when measurement was stamped in the body IMU and leg encoder's clocks.

On the other hand, we do assume the leg encoders are all synchronized, which is a reasonable and common assumption

in practice in order to synchronize all actuators. However, we do need to determine the lengths of the three-link segments $\mathbf{x}_{LI} = [l_1 \; l_2 \; l_3]^\top$, as they are required to capitalize the leg kinematics (2) and may undergo variations due to contact-induced deformation during operations (e.g., walking).

IV. Determining Kinematic Parameters

As discussed in the previous section, determining the legged kinematics (2) requires to identify the parameters including the leg-body transformation $\{{}^L\mathbf{p}_I, {}^L_I\mathbf{R}\}$ and time offset ${}^I t_L$ and the link lengths $\mathbf{x}_{LI}$. To this end, in analogy to the full calibration of multi-visual-inertial systems whose extrinsics is akin to the leg-body transformation and baselines between IMUs (or cameras) are parallel to the leg links, 3-axis fully excited motion is needed to ensure observability of these kinematic parameters. Although the standard walking or running gaits of a legged robot would produce such fully excited motion, motion along certain axis (e.g., roll and pitch if walking on flat ground) might not be significant enough to guarantee fast convergence, and the time or space constraints imposed by these gaits may not be possible in practice. Bearing that in mind, we advocate to perform an *in-place dance* motion[1] for a short period of time (e.g., 25 seconds in our experiments) where the toes remain static on ground without consuming much space (see Fig. 1), which would generate adequate motions for identifying our kinematic parameters. Interestingly, during this in-place dance, the static toes of the legged robot can serve as "landmarks" as in SLAM for our kinematics identification to bound estimation errors, while avoiding the notorious contact detection inaccuracies.

Evidently, it is essential to be able to efficiently and accurately track 3D motion of the robot's body (or floating base) in order to determine the kinematic parameters. To this end, we employ visual-inertial sensors which are typically available onboard robots and leverage our prior work OpenVINS [6] which is an efficient MSCKF-based VINS estimator and offers accurate 3D motion tracking of the sensor platform. As such, conceptually, we can treat the VINS as 3D "odometry" and employ the legged kinematic constraints (2) as 3D position measurements to build a 3D feature-based SLAM system to estimate the kinematic parameters. In particular, the state vector of the proposed estimator includes the kinematic parameters along with the standard navigation states:

$$\mathbf{x}_k = \begin{bmatrix}\mathbf{x}_{I_k}^\top & \mathbf{x}_{C_k}^\top & \mathbf{x}_T^\top & \mathbf{x}_\Theta^\top\end{bmatrix}^\top \tag{8}$$

$$\mathbf{x}_{I_k} = \begin{bmatrix}{}^{I_k}_G\bar{q}^\top & {}^G\mathbf{p}_{I_k}^\top & {}^G\mathbf{v}_{I_k}^\top & \mathbf{b}_g^\top & \mathbf{b}_a^\top\end{bmatrix}^\top \tag{9}$$

$$\mathbf{x}_{C_k} = \begin{bmatrix}{}^{I_{k-1}}_G\bar{q}^\top & {}^G\mathbf{p}_{I_{k-1}}^\top & \cdots & {}^{I_{k-n}}_G\bar{q}^\top & {}^G\mathbf{p}_{I_{k-n}}^\top\end{bmatrix}^\top \tag{10}$$

$$\mathbf{x}_T = \begin{bmatrix}{}^G\mathbf{p}_{T_1}^\top & \cdots & {}^G\mathbf{p}_{T_w}^\top\end{bmatrix}^\top \tag{11}$$

$$\mathbf{x}_{L_i} = \begin{bmatrix}{}^{L_i}_I\bar{q}^\top & {}^{L_i}\mathbf{p}_I^\top & l_1^i & l_2^i & l_3^i\end{bmatrix}^\top \tag{12}$$

$$\mathbf{x}_\Theta = \begin{bmatrix}\mathbf{x}_{L_1}^\top & \cdots & \mathbf{x}_{L_w}^\top & {}^L t_I\end{bmatrix}^\top \tag{13}$$

where $\mathbf{x}_{I_k}$ is the current body/IMU state [27], $\mathbf{x}_{C_k}$ are the n historical cloned poses in the global frame $\{G\}$, $\mathbf{x}_T$ are the w static toe landmark positions, $\mathbf{x}_{L_i}$ are the IMU-LEG spatial extrinsic and link length parameters for i^{th} leg, and $\mathbf{x}_\Theta$ are the kinematic parameters of w legs along with a common temporal time offset between IMU and leg encoders.[2]

[1] For example, see: https://www.youtube.com/watch?v=vLSq6wDJm7s

A. VINS-based Motion Tracking of Floating Base

As a common practice, we assume the legged robot system is represented by the IMU states on the robot's body or floating base, which the locomotion control is applied to. We build upon OpenVINS [6] to efficiently track its motion. Specifically, the IMU state (9) evolves over time with the IMU kinematics, using incoming linear acceleration ${}^I\mathbf{a}_k$, and angular velocity ${}^I\boldsymbol{\omega}_k$, which is given by (see [29]):

$$\mathbf{x}_{I_{k+1}} = \mathbf{f}_I(\mathbf{x}_{I_k}, {}^I\mathbf{a}_k, {}^I\boldsymbol{\omega}_k, \mathbf{n}_I) \tag{14}$$

where $\mathbf{n}_I = [\mathbf{n}_a^\top \; \mathbf{n}_g^\top \; \mathbf{n}_{wa}^\top \; \mathbf{n}_{wg}^\top]^\top$ contains the zero-mean white Gaussian noises and random walks of the biases. We detect and track static environmental features over images with optical flow. The bearing measurement $\mathbf{z}_{C_k}$ of a feature $\mathbf{p}_f$ at timestep k is the perspective projection of its 3D global position ${}^G\mathbf{p}_f$ onto the image plane:

$$\mathbf{z}_{C_k} = \mathbf{h}_c(\mathbf{x}_{C_k}, {}^{C_k}\mathbf{p}_f) + \mathbf{n}_{C_k} \tag{15}$$

$${}^{C_k}\mathbf{p}_f = {}^C_I\mathbf{R}\,{}^{I_k}_G\mathbf{R}({}^G\mathbf{p}_f - {}^G\mathbf{p}_{I_k}) + {}^C\mathbf{p}_I \tag{16}$$

where $\{{}^C_I\mathbf{R}, {}^C\mathbf{p}_I\}$ is the IMU-camera extrinsic transformation which is assumed to be known otherwise can be calibrated either online or offline [4], and $\mathbf{n}_{C_k}$ is the white Gaussian measurement noise. We now employ the MSCKF to efficiently update the state with these visual measurements [27].

B. Estimating Legged Kinematics

Provided the motion tracking (i.e., "odometry") of the robot's body as described in the preceding section and given the legged kinematic constraint in Eq. (2), one may simply build a decoupled SLAM estimator by including the kinematic parameters and the static toes (as landmarks) as part of the state vector along with the robot body/IMU states in order to identify the kinematics. However, as decoupling the legged kinematics from the body's motion tracking may cause information loss, we propose to tightly couple the kinematic parameters into VINS so that the body's motion tracking can also benefit from the legged kinematics.

Specifically, the forward legged kinematics (2) reveals the relationship of the toe position in the body frame, ${}^I\mathbf{p}_{T_k}$, with the joint encoder measurements and the kinematic parameters (which are included in our state vector (8)). On the other hand, we can also represent the toe position in terms of the body pose and the global toe position (both of which are part of the state vector (8)) as follows:

$${}^I\mathbf{p}_{T_k} = {}^{I_k}_G\mathbf{R}({}^G\mathbf{p}_T - {}^G\mathbf{p}_{I_k}) \tag{17}$$

To infer a legged kinematic measurement model that can be used in the proposed tightly-coupled MSCKF update,

[2] ${}^{I_k}_G\bar{q}$ is the JPL unit quaternion [28] corresponding to the rotation ${}^{I_k}_G\mathbf{R}$ from $\{G\}$ to $\{I\}$, ${}^G\mathbf{p}_{I_k}$ and ${}^G\mathbf{v}_{I_k}$ are the position and velocity of $\{I\}$ in $\{G\}$, and $\mathbf{b}_g$ and $\mathbf{b}_a$ are the biases of the gyroscope and accelerometer, respectively.

we equate (2) and (17) and build the following implicit measurement (note that we drop off the time index for simplicity):

$$\mathbf{z}_\ell := \mathbf{0} = {}^I_G\mathbf{R}\,{}^G\mathbf{p}_T - {}^I_G\mathbf{R}\,{}^G\mathbf{p}_I - {}^I\mathbf{p}_T =: \mathbf{h}_\ell(\mathbf{x}, \boldsymbol{\theta}_m) \quad (18)$$

Where $\boldsymbol{\theta}_m$ denotes the stacked joint encoder measurements which are corrupted by zero-mean white Gaussian noise $\mathbf{n}_\ell = [n_{\theta_1}\ n_{\theta_2}\ n_{\theta_3}]^\top$ (see (1)). The residual of this inferred legged measurement (18) is obtained by linearizing with respect to the state $\mathbf{x}$[3] and encoder noise $\mathbf{n}_\ell$ as:

$$\mathbf{r}_\ell = \mathbf{z}_\ell - \hat{\mathbf{z}}_\ell = -\hat{\mathbf{z}}_\ell = \mathbf{H}_\mathbf{x}\tilde{\mathbf{x}} + \mathbf{G}_\mathbf{n}\mathbf{n}_\ell \quad (19)$$

where the two Jacobians are respectively defined by $\mathbf{H}_\mathbf{x} = \begin{bmatrix} \frac{\partial \mathbf{h}_\ell}{\partial {}^I_G\boldsymbol{\theta}} & \frac{\partial \mathbf{h}_\ell}{\partial {}^G\mathbf{p}_I} & \frac{\partial \mathbf{h}_\ell}{\partial {}^G\mathbf{p}_T} & \frac{\partial \mathbf{h}_\ell}{\partial {}^I t_L} & \frac{\partial \mathbf{h}_\ell}{\partial {}^L_I\boldsymbol{\theta}} & \frac{\partial \mathbf{h}_\ell}{\partial {}^L\mathbf{p}_I} & \frac{\partial \mathbf{h}_\ell}{\partial l_1} & \frac{\partial \mathbf{h}_\ell}{\partial l_2} & \frac{\partial \mathbf{h}_\ell}{\partial l_3} \end{bmatrix}$ and $\mathbf{G}_n = \begin{bmatrix} \frac{\partial \mathbf{h}_\ell}{\partial n_{\theta_A}} & \frac{\partial \mathbf{h}_\ell}{\partial n_{\theta_H}} & \frac{\partial \mathbf{h}_\ell}{\partial n_{\theta_K}} \end{bmatrix}$, and are given by:

$$\frac{\partial \mathbf{h}_\ell}{\partial {}^I_G\boldsymbol{\theta}} = \lfloor {}^I_G\hat{\mathbf{R}}({}^G\hat{\mathbf{p}}_T - {}^G\hat{\mathbf{p}}_I)\rfloor\ ,\ \frac{\partial \mathbf{h}_\ell}{\partial {}^G\mathbf{p}_I} = -{}^I_G\hat{\mathbf{R}} \quad (20)$$

$$\frac{\partial \mathbf{h}_\ell}{\partial {}^G\mathbf{p}_T} = {}^I_G\hat{\mathbf{R}}\ ,\ \frac{\partial \mathbf{h}_\ell}{\partial l_3} = -{}^L_I\hat{\mathbf{R}}^\top {}^L_A\mathbf{R}\,{}^A_H\mathbf{R}\,{}^H_K\mathbf{R}\mathbf{k}_3 \quad (21)$$

$$\frac{\partial \mathbf{h}_\ell}{\partial {}^I t_L} = -\lfloor {}^I_G\hat{\mathbf{R}}({}^G\hat{\mathbf{p}}_T - {}^G\hat{\mathbf{p}}_I)\rfloor {}^I\boldsymbol{\omega} + {}^I_G\hat{\mathbf{R}}\,{}^G\mathbf{v}_I \quad (22)$$

$$\frac{\partial \mathbf{h}_\ell}{\partial {}^L_I\boldsymbol{\theta}} = -{}^L_I\hat{\mathbf{R}}^\top \lfloor ({}^L\hat{\mathbf{p}}_I - {}^L\mathbf{p}_T)\rfloor\ ,\ \frac{\partial \mathbf{h}_\ell}{\partial {}^L\mathbf{p}_I} = {}^L_I\hat{\mathbf{R}}^\top \quad (23)$$

$$\frac{\partial \mathbf{h}_\ell}{\partial l_1} = -{}^L_I\hat{\mathbf{R}}^\top {}^L_A\mathbf{R}\mathbf{k}_2\ ,\ \frac{\partial \mathbf{h}_\ell}{\partial l_2} = -{}^L_I\hat{\mathbf{R}}^\top {}^L_A\mathbf{R}\,{}^A_H\mathbf{R}\mathbf{k}_3 \quad (24)$$

$$\frac{\partial \mathbf{h}_\ell}{\partial n_{\theta_A}} = -{}^L_I\hat{\mathbf{R}}^\top {}^L_A\mathbf{R}\lfloor {}^A\mathbf{p}_T\rfloor \mathbf{J}_\mathbf{r}(\theta_{m,A}\mathbf{k}_A)\mathbf{k}_A \quad (25)$$

$$\frac{\partial \mathbf{h}_\ell}{\partial n_{\theta_H}} = -{}^L_I\hat{\mathbf{R}}^\top {}^L_A\mathbf{R}\,{}^A_H\mathbf{R}\lfloor {}^H\mathbf{p}_T\rfloor \mathbf{J}_\mathbf{r}(\theta_{m,H}\mathbf{k}_H)\ \mathbf{k}_H \quad (26)$$

$$\frac{\partial \mathbf{h}_\ell}{\partial n_{\theta_K}} = -{}^L_I\hat{\mathbf{R}}^\top {}^L_A\mathbf{R}\,{}^A_H\mathbf{R}\,{}^H_K\mathbf{R}\lfloor {}^K\mathbf{p}_T\rfloor \mathbf{J}_\mathbf{r}(\theta_{m,K}\mathbf{k}_K)\mathbf{k}_K \quad (27)$$

$${}^H\mathbf{p}_T = {}^H\mathbf{p}_K + {}^H_K\mathbf{R}\,{}^K\mathbf{p}_T\ ,\ {}^A\mathbf{p}_T = {}^A\mathbf{p}_H + {}^A_H\mathbf{R}\,{}^H\mathbf{p}_T$$

where $\lfloor\cdot\rfloor$ is the skew-symmetric matrix, ${}^I_G\boldsymbol{\theta}$ is $\mathbb{R}^3$ representation of corresponding rotation ${}^I_G\mathbf{R}$ in $SO(3)$. These Jacobians are critical to ensure accurate and consistent estimation, and their detailed derivations can be found in our supplementary technical report [30]. A crucial aspect of this linearization is to properly compute Jacobian $\mathbf{G}_\mathbf{n}$ with respect to the encoder noise. It is important to note that when evaluating the above linearization, we employ the first estimates Jacobian (FEJ) methodology [31], of the toe landmarks ${}^G\mathbf{p}_T$, to ensure estimation consistency.

C. Contact Detection via Consistent Estimation

In general scenarios, at this point, we are ready to use the above legged measurement residual (19), together with the visual measurement residual as in VINS (see (15)), to perform tightly-coupled EKF update of both the body's motion states and the leg's kinematic parameters. In the following, we take special care for toe landmarks when a legged robot performs different motion gaits, which are of practical significance.

[3]Throughout this paper $\hat{\mathbf{x}}$ is used to denote the estimate of a random variable $\mathbf{x}$, while $\tilde{\mathbf{x}} = \mathbf{x} \boxminus \hat{\mathbf{x}}$ is the error of this estimate. We define the orientation error quaternion, $\delta\theta$, as $\bar{\mathbf{q}} \otimes \hat{\bar{\mathbf{q}}}^{-1} \simeq \begin{bmatrix} \frac{1}{2}\delta\theta^\top & 1 \end{bmatrix}^\top$ [28].

TABLE I: Simulation parameters and prior single standard deviations were drawn from perturbations of measurements and initial states.

Parameter	Value	Parameter	Value
Cam Freq. (Hz)	10	IMU Freq. (Hz)	400
Leg Freq. (Hz)	50	Num. Clones	11
Pixel Proj. (px)	1	Leg White Noise (m)	2.0e-02
Gyro. White Noise	5.4e-04	Gyro. Rand. Walk	1.6e-05
Accel. White Noise	7.3e-03	Accel. Rand. Walk	6.6e-04
Num. Legs	4	Leg Toff. Ptrb. (sec)	8.0e-03
Leg Ext (Ori). Ptrb. (rad)	0.015	Leg Ext (Pos). Ptrb. (m)	3.0e-02

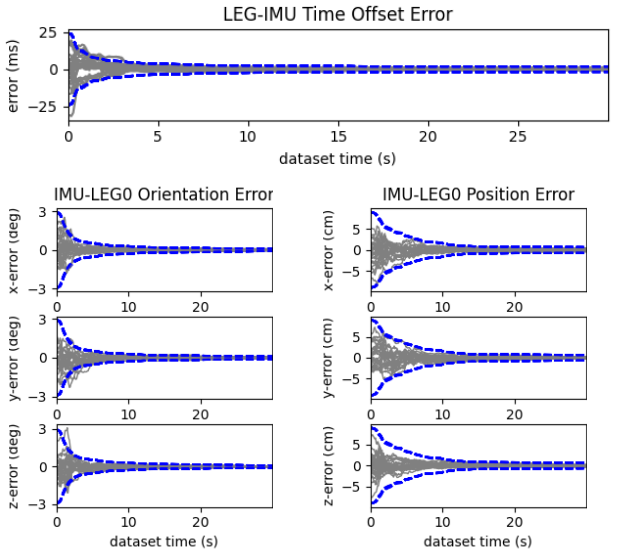

Fig. 3: Extrinsic calibration errors (solid) and 3σ bound (dotted) for 25 different runs under random motion. Each solid line denotes a run with a different realization of the measurement noise and the initial values.

Note that the above EKF update with the leg measurements is only valid while the toe is in contact at the same point; otherwise, it could hurt our estimation performance. This reveals the fact that robust and accurate contact detection during walking is crucial for fusing leg information, with various contact detection methods being investigated [32], [33]. However, we take advantage of the proper legged kinematic model and consistent covariance estimates available from our MSCKF, and adopt the Mahalanobis distance test. Specifically, we perform the following threshold check to see if the toe is in the same position:

$$\mathbf{r}_{\ell_k}^\top (\mathbf{H}_{\mathbf{x},\mathbf{k}}\ \mathbf{P}_{k|k}\ \mathbf{H}_{\mathbf{x},\mathbf{k}}{}^\top + \mathbf{G}_{\mathbf{n},\mathbf{k}}\mathbf{R}'\mathbf{G}_{\mathbf{n},\mathbf{k}}{}^\top)^{-1}\ \mathbf{r}_{\ell_k} < \chi^2$$

where $\mathbf{P}_{k|k}$ is the covariance of the augmented state, $\mathbf{H}_{\mathbf{x},\mathbf{k}}, \mathbf{G}_{\mathbf{n},\mathbf{k}}$ consists of the Jacobians (19), $\mathbf{R}'$ consists of the leg encoder noise variance, and χ is the threshold for the test and is set at 95%. When the test fails, we consider the toe is lifted from the ground and marginalize ${}^G\mathbf{p}_T$ from the state. The measurement will be used to update the state as a contact constraint via the legged kinematics.

V. Simulation Results

To verify the proposed online kinematic determination, we extended the visual-inertial simulator based on OpenVINS [6] to simulate quadruped motion and generate leg

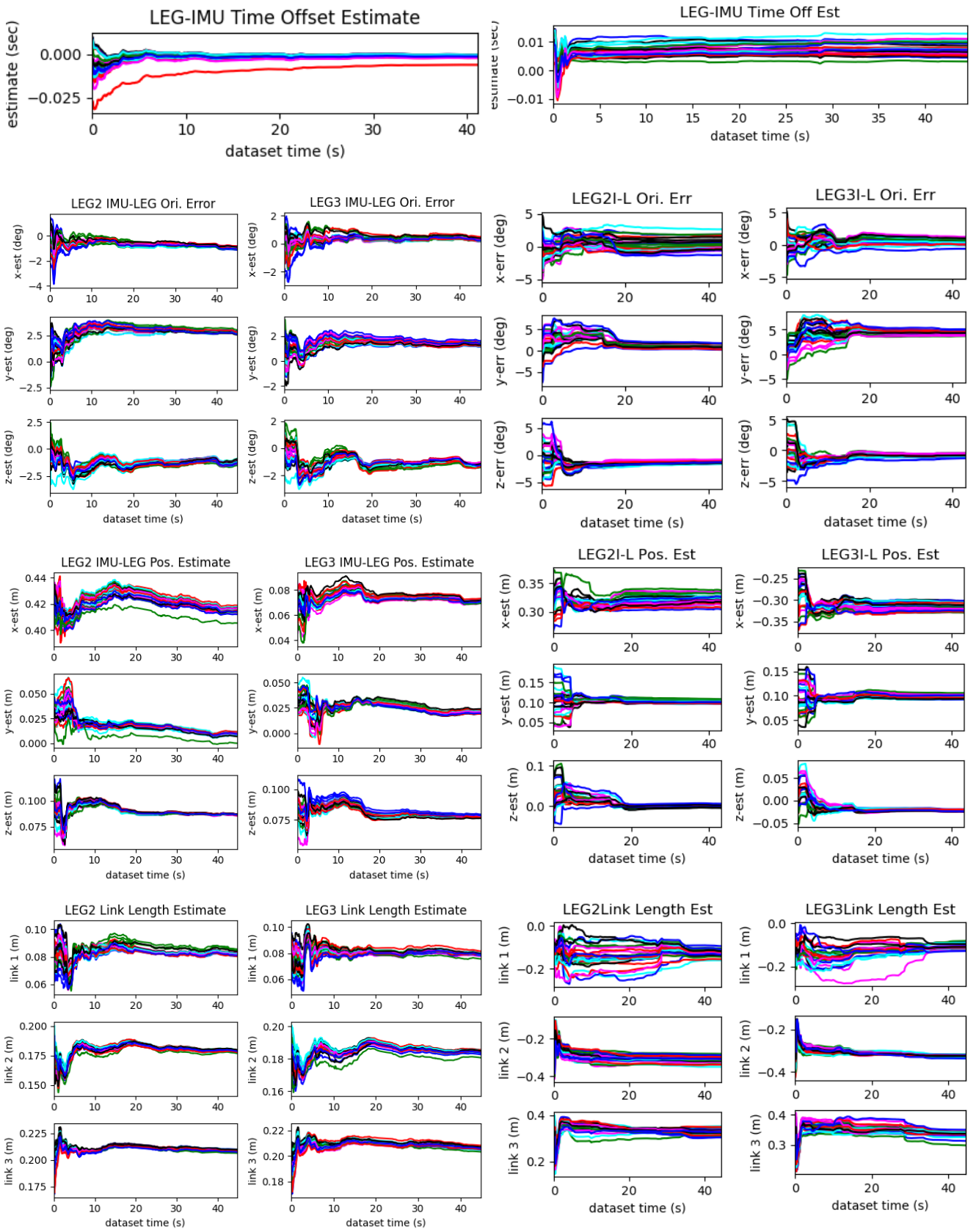

Fig. 4: Estimation results of the legged kinematic parameters for the *Dance* dataset: (left) *Jueying Lite 2*, and (right) *Ghost Vision 60*. Each color denotes runs with different initial guesses of the parameters (a total of 25 runs).

TABLE II: Parameter repeatability over 25 runs with different initial conditions (see Tab. III for initial distributions) for *Jueying Lite 2* and *Ghost Vision 60* quadruped. All values reported are a single standard deviation after a 30-second dance for "dance" dataset and a 30-second dance followed by two minutes of walk in "dance_walk" dataset.

Datasets	Leg0 Rot (deg)	Leg0 Pos (cm)	Leg0 Intrinsic (cm)	Toff (ms)
jye_dance 1	$0.18 \pm 0.20, 0.87 \pm 0.11, -1.10 \pm 0.22$	$7.74 \pm 0.11, -10.02 \pm 0.13, 7.89 \pm 0.07$	$7.30 \pm 0.06, 19.00 \pm 0.09, 21.00 \pm 0.08$	-1.35 ± 1.55
jye_dance 2	$1.25 \pm 0.11, 1.01 \pm 0.09, -0.34 \pm 0.16$	$8.23 \pm 0.09, -10.01 \pm 0.16, 8.18 \pm 0.08$	$7.00 \pm 0.10, 19.00 \pm 0.09, 22.00 \pm 0.14$	-2.0 ± 0.77
jye_dance 3	$0.09 \pm 0.05, 0.94 \pm 0.07, -1.58 \pm 0.06$	$8.97 \pm 0.08, -10.29 \pm 0.15, 8.32 \pm 0.06$	$8.33 \pm 0.14, 19.00 \pm 0.06, 22.00 \pm 0.06$	-1.21 ± 1.14
jye_dance_walk 1	$-1.19 \pm 0.31, -2.9 \pm 0.12, -1.18 \pm 0.29$	$9.82 \pm 0.33, -10.44 \pm 0.26, 6.86 \pm 0.27$	$10.0 \pm 0.30, 21.0 \pm 0.16, 24.0 \pm 0.15$	-0.50 ± 0.64
jye_dance_walk 2	$-0.54 \pm 0.32, -3.93 \pm 0.19, -1.21 \pm 0.23$	$9.40 \pm 0.23, -9.44 \pm 0.23, 5.50 \pm 0.17$	$9.20 \pm 0.18, 20.09 \pm 0.20, 23.00 \pm 0.28$	0.09 ± 0.73
Average	$-0.04 \pm 0.85, -0.79 \pm 2.17, -1.07 \pm 0.46$	$8.83 \pm 0.78, -10.04 \pm 0.39, 7.36 \pm 1.07$	$8.20 \pm 1.30, 20.01 \pm 0.81, 22.01 \pm 0.90$	-1.05 ± 1.22
ghost_dance 1	$2.73 \pm 0.92, 3.93 \pm 0.94, -0.20 \pm 0.30$	$-29.36 \pm 0.44, -10.64 \pm 0.36, -1.13 \pm 0.48$	$9.74 \pm 2.21, -31.56 \pm 0.73, 34.17 \pm 0.96$	7.84 ± 2.36
ghost_dance 2	$2.19 \pm 0.85, 4.68 \pm 0.49, -0.59 \pm 0.17$	$-31.24 \pm 1.16, -10.82 \pm 0.44, -2.09 \pm 0.24$	$8.25 \pm 1.18, -34.30 \pm 1.52, 33.14 \pm 1.31$	9.08 ± 2.75
ghost_dance 3	$0.97 \pm 0.90, 3.31 \pm 0.28, -0.93 \pm 0.38$	$-30.74 \pm 1.36, -10.64 \pm 0.42, -1.26 \pm 0.28$	$9.77 \pm 0.94, -34.14 \pm 1.47, 33.19 \pm 1.42$	0.54 ± 4.46
ghost_dance_walk 1	$1.05 \pm 2.00, 2.46 \pm 1.67, -1.27 \pm 0.42$	$-28.58 \pm 1.66, -10.29 \pm 0.58, -1.02 \pm 0.97$	$8.96 \pm 1.46, -32.02 \pm 1.64, 30.56 \pm 1.89$	11.60 ± 2.63
ghost_dance_walk 2	$3.04 \pm 3.34, 2.68 \pm 1.32, -1.00 \pm 0.45$	$-29.90 \pm 1.47, -10.75 \pm 0.56, -2.55 \pm 0.81$	$6.75 \pm 3.99, -32.27 \pm 1.31, 32.75 \pm 2.03$	8.50 ± 1.13
Average	$2.00 \pm 2.03, 3.41 \pm 1.33, -0.80 \pm 0.51$	$-29.97 \pm 1.58, -10.63 \pm 0.50, -1.15 \pm 0.85$	$8.69 \pm 2.48, -32.86 \pm 1.77, 32.76 \pm 1.96$	7.51 ± 4.67

TABLE III: Real-world experiment parameters and prior standard deviations which perturbations of measurements and initial states for *Jueying Lite 2* and *Ghost Vision 60*.

Jueying Lite 2 Parameter	Value	Ghost Vision 60 Parameter	Value
Pixel Proj. (px)	1	Pixel Proj. (px)	1
Link Length. Ptrb. (m)	0.010	Link Length. Ptrb. (m)	0.04
Leg Ext (Ori). Ptrb. (rad)	0.017	Leg Ext (Ori). Ptrb. (rad)	0.043
Leg encoder Noise (rad)	0.005	Leg encoder Noise (rad)	0.006
Leg Toff. Ptrb. (sec)	0.003	Leg Toff. Ptrb. (sec)	0.004
Leg Ext (Pos). Ptrb. (m)	0.010	Leg Ext (Pos). Ptrb. (m)	0.03

TABLE IV: Absolute Trajectory Error (ATE) (deg/m) of each algorithm on *Jueying Lite 2* and *Ghost Vision 60*. The best results are highlighted in bold font.

Datasets	Leg-VINS wo. Calib	Leg-VINS w. Calib
jye_dance_walk 1	6.596 / **0.387**	**5.466** / 0.437
jye_dance_walk 2	7.258 / 0.681	**6.314** / **0.495**
jye_dance_walk 3	13.317 / 1.158	**10.497** / **0.822**
jye_dance_walk 4	6.736 / 0.550	**6.509** / **0.433**
Average	8.476 / 0.694	**7.196** / **0.547**
ghost_dance_walk 1	8.867 / **0.283**	**7.186** / 0.292
ghost_dance_walk 2	12.045 / 0.583	**7.929** / **0.455**
ghost_dance_walk 3	11.981 / 0.523	**8.144** / **0.515**
Average	10.96 / 0.463	**7.753** / **0.420**

measurements. Note that we here focus on evaluating only the kinematic parameters of the spatial rigid transformation between the robot's body and leg and the time offset between the body IMU and leg encoders, which are body-leg spatiotemporal parameters. During the simulation, the position of the toe in the leg frame, ${}^{L}\mathbf{p}_{T_k}$, is directly simulated with additive white noise perturbations. We include our initial perturbation of spatiotemporal parameters to get the kinematic measurements as in (2). We performed 25 Monte-Carlo runs with different initial perturbations, see Tab. I, and have shown a single representative leg result in Fig. 3. It is clear that the proposed method is able to accurately recover the spatiotemporal parameters within 20 seconds, and the estimated uncertainty captures the true error distribution.

VI. Experimental Results

We have also evaluated the proposed method on two quadruped robots, *Jueying Lite 2* and *Ghost Vision 60*, which are equipped with a stereo camera and an IMU, along with four 3-link chain legs (see Fig. 1). Two different motion scenarios are captured: (1) *dance* in which toe remains in contact with the ground for the whole duration and enables evaluation of the calibration repeatability, and (2) *dance_walk* where after 10-15 seconds of dancing motion, the robot walks within a motion capture room so that we can quantify the impact of good and bad calibration of localization performance after parameter identification. For Dancing motion, we provided different combinations of roll, pitch, and yaw commands while keeping their feet stationary at all times. There is no particular dancing motion requirement; any combination that thoroughly excites all the joints and body will suffice. All *dance* sequences are about 25-40 seconds, and *dance_walk* sequences are about 2 minutes; their trajectory plots and additional results can be found in our supplementary material [30]. The datasets for the two robots are distinguished as *jye_dance* and *ghost_dance* for the *dance* datasets, and *jye_dance_walk* and *ghost_dance_walk* for *dance_walk* datasets.

We first evaluated on the *dance* sequences to see the performance without being affected by the contact detection problem. The kinematic results of two legs are shown in Fig. 4, while results for the remaining legs can be found in the supplementary material [30]. For each run, we perturb the manufacturer-provided calibration (URDF) by the values in Tab. III, and we can see that most parameters quickly converged after 25 seconds of calibration start. The converged values are near our best estimate and have a low variance compared to prior perturbation variance, indicating the proposed system's fast and reliable identification performance. To quantify the repeatability, we look at the variance of the final estimated value in Tab. II for a representative leg on each dataset. The *dance_walk* datasets have higher variance, which we attribute to evaluation after walking, which is affected by the contact detection accuracy. We also calculate the average value for all the parameters and their variances across all the datasets, and similar results can be seen for *Ghost Vision 60* for both types of datasets. This shows that we are able to calibrate with good repeatability using the proposed estimator.

We further evaluate the estimation performance on all sequences (*dance_walk*) in terms of localization accuracy by comparing estimation with and without calibrating legged kinematic parameters (tagged Leg-VINS w. Calib and Leg-VINS wo. Calib, respectively). Note that we set the same initial kinematic values for all compared methods. The

Absolute Trajectory Errors (ATE) [34] are shown in Tab. IV for *Jueying Lite 2*, the orientation error reduces by 15% and position error reduces by 20%, for *Ghost Vision 60* the orientation error reduces by 30% and position error reduces by 20% for a two-minute walking datasets which clearly shows that the proposed Leg-VINS w. Calib improves the localization accuracy by properly incorporating the legged kinematic constraints as compared with Leg-VINS wo. Calib. This reconfirms that localization accuracy can be improved through the proposed online kinematic determination.

VII. Conclusions and Future Work

In this paper, we have designed a user-friendly procedure featuring in-place dance motions that quickly determine the legged kinematics, which is crucial for legged locomotion and state estimation. We have developed a tightly-coupled efficient MSCKF-based estimator to determine the kinematic parameters simultaneously modeling and fusing the leg kinematic constraints which improves the state estimation and consequently locomotion control. Extensive evaluations were performed to demonstrate that the proposed method shows the repeatability and convergence of these kinematic parameters and improve estimation accuracy.

References

[1] M. Hutter, C. Gehring, D. Jud, A. Lauber, C. D. Bellicoso, V. Tsounis, J. Hwangbo, K. Bodie, P. Fankhauser, M. Bloesch *et al.*, "Anymal-a highly mobile and dynamic quadrupedal robot," in *IEEE/RSJ international conference on intelligent robots and systems (IROS)*, 2016, pp. 38–44.

[2] A. Meduri, P. Shah, J. Viereck, M. Khadiv, I. Havoutis, and L. Righetti, "Biconmp: A nonlinear model predictive control framework for whole body motion planning," *IEEE Transactions on Robotics*, vol. 39, no. 2, pp. 905–922, 2023.

[3] M. Camurri, M. Ramezani, S. Nobili, and M. Fallon, "Pronto: A multi-sensor state estimator for legged robots in real-world scenarios," *Frontiers in Robotics and AI*, vol. 7, p. 68, 2020.

[4] Y. Yang, P. Geneva, X. Zuo, and G. Huang, "Online self-calibration for visual-inertial navigation: Models, analysis, and degeneracy," *IEEE Transactions on Robotics*, vol. 39, no. 5, pp. 3479–3498, 2023.

[5] M. Maravgakis, D.-E. Argiropoulos, S. Piperakis, and P. Trahanias, "Probabilistic contact state estimation for legged robots using inertial information," in *IEEE International Conference on Robotics and Automation (ICRA)*, vol. 2, no. 2, 2023, pp. 1023–1030.

[6] P. Geneva, K. Eckenhoff, W. Lee, Y. Yang, and G. Huang, "OpenVINS: a research platform for visual-inertial estimation," in *Proc. of the IEEE International Conference on Robotics and Automation*, Paris, France, 2020, pp. 4666–4672. [Online]. Available: https://github.com/rpng/open_vins

[7] Z. Roth, B. Mooring, and B. Ravani, "An overview of robot calibration," *IEEE Journal on Robotics and Automation*, vol. 3, no. 5, pp. 377–385, 1987.

[8] K. H. Strobl and G. Hirzinger, "Optimal hand-eye calibration," in *IEEE/RSJ international conference on intelligent robots and systems*, 2006, pp. 4647–4653.

[9] K. Nickels, E. Huber, and M. DiCicco, "Hand-eye calibratilon using active vision," in *2007 IEEE Aerospace Conference*, 2007, pp. 1–9.

[10] O. Birbach, U. Frese, and B. Bäuml, "Rapid calibration of a multi-sensorial humanoid's upper body: An automatic and self-contained approach," *The International Journal of Robotics Research*, vol. 34, no. 4-5, pp. 420–436, 2015.

[11] T. Kastner, T. Röfer, and T. Laue, "Automatic robot calibration for the nao," in *RoboCup 2014: Robot World Cup XVIII 18*. Springer, 2015, pp. 233–244.

[12] F. Blöchliger, M. Blösch, P. Fankhauser, M. Hutter, and R. Siegwart, "Foot-eye calibration of legged robot kinematics," in *Advances in Cooperative Robotics*. World Scientific, 2017, pp. 420–427.

[13] M. Bloesch, M. Hutter, C. Gehring, M. A. Hoepflinger, and R. Siegwart, "Kinematic batch calibration for legged robots," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2013, pp. 2542–2547.

[14] S. Yang, H. Choset, and Z. Manchester, "Online kinematic calibration for legged robots," *IEEE Robotics and Automation Letters*, vol. 7, no. 3, pp. 8178–8185, 2022.

[15] S. Yang, Z. Zhang, Z. Fu, and Z. Manchester, "Cerberus: Low-drift visual-inertial-leg odometry for agile locomotion," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2023, pp. 4193–4199.

[16] G. Huang, "Visual-inertial navigation: A concise review," in *Proc. International Conference on Robotics and Automation*, Montreal, Canada, May 2019, pp. 9572–9582.

[17] C. Chen, Y. Yang, P. Geneva, W. Lee, and G. Huang, "Visual-inertial-aided online mav system identification," in *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2022, pp. 6277–6284.

[18] W. Lee, K. Eckenhoff, Y. Yang, P. Geneva, and G. Huang, "Visual-inertial-wheel odometry with online calibration," in *Proc. of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, Las Vegas, NV, 2020, pp. 4559–4566.

[19] M. Bloesch, M. Hutter, M. A. Hoepflinger, S. Leutenegger, C. Gehring, C. D. Remy, and R. Siegwart, "State estimation for legged robots-consistent fusion of leg kinematics and imu," *Robotics*, 2013.

[20] R. Hartley, M. Ghaffari, R. M. Eustice, and J. W. Grizzle, "Contact-aided invariant extended kalman filtering for robot state estimation," *The International Journal of Robotics Research*, vol. 39, no. 4, pp. 402–430, 2020.

[21] D. Wisth, M. Camurri, and M. Fallon, "Robust legged robot state estimation using factor graph optimization," *IEEE Robotics and Automation Letters (RA-L)*, vol. 4, no. 4, pp. 4507–4514, 2019.

[22] 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 *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2018, pp. 3783–3790.

[23] D. Wisth, M. Camurri, and M. Fallon, "Vilens: Visual, inertial, lidar, and leg odometry for all-terrain legged robots," *IEEE Transactions on Robotics*, vol. 39, no. 1, pp. 309–326, 2022.

[24] Y. Kim, B. Yu, E. M. Lee, J.-h. Kim, H.-w. Park, and H. Myung, "Step: State estimator for legged robots using a preintegrated foot velocity factor," *IEEE Robotics and Automation Letters (RA-L)*, vol. 7, no. 2, pp. 4456–4463, 2022.

[25] C. Mastalli, R. Budhiraja, W. Merkt, G. Saurel, B. Hammoud, M. Naveau, J. Carpentier, L. Righetti, S. Vijayakumar, and N. Mansard, "Crocoddyl: An efficient and versatile framework for multi-contact optimal control," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2020, pp. 2536–2542.

[26] G. Chirikjian, *Stochastic Models, Information Theory, and Lie Groups, Volume 2: Analytic Methods and Modern Applications*. Springer Science & Business Media, 2011, vol. 2.

[27] A. I. Mourikis and S. I. Roumeliotis, "A multi-state constraint Kalman filter for vision-aided inertial navigation," in *Proceedings of the IEEE International Conference on Robotics and Automation*, Rome, Italy, 2007, pp. 3565–3572.

[28] N. Trawny and S. I. Roumeliotis, "Indirect Kalman filter for 3D attitude estimation," University of Minnesota, Dept. of Comp. Sci. & Eng., Tech. Rep., Mar. 2005.

[29] A. B. Chatfield, *Fundamentals of high accuracy inertial navigation*. Aiaa, 1997, vol. 174.

[30] C. Burgul, W. Lee, P. Geneva, and G. Huang, "Supplementary material: Visual-inertial-aided kinematic calibration of legged robot," University of Delaware, Dept. of Mechanical Eng., Tech. Rep., 2024, available: http://udel.edu/~ghuang/papers/tr_legged.pdf.

[31] G. Huang, A. I. Mourikis, and S. I. Roumeliotis, "Observability-based rules for designing consistent EKF SLAM estimators," *International Journal of Robotics Research*, vol. 29, no. 5, pp. 502–528, 2010.

[32] M. Camurri, M. Fallon, S. Bazeille, A. Radulescu, V. Barasuol, D. G. Caldwell, and C. Semini, "Probabilistic contact estimation and impact detection for state estimation of quadruped robots," *IEEE Robotics and Automation Letters (RA-L)*, vol. 2, no. 2, pp. 1023–1030, 2017.

[33] Q. Liu, B. Yuan, and Y. Wang, "Online learning for foot contact detection of legged robot based on data stream clustering," *Frontiers in Bioengineering and Biotechnology*, vol. 9, p. 771415, 2022.

[34] Z. Zhang and D. Scaramuzza, "A tutorial on quantitative trajectory evaluation for visual (-inertial) odometry," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2018, pp. 7244–7251.