Legged Robot State Estimation in Slippery Environments Using Invariant Extended Kalman Filter with Velocity Update

Sangli Teng, Mark Wilfried Mueller, and Koushil Sreenath

Abstract— This paper proposes a state estimator for legged robots operating in slippery environments. An Invariant Extended Kalman Filter (InEKF) is implemented to fuse inertial and velocity measurements from a tracking camera and leg kinematic constraints. The misalignment between the camera and the robot-frame is also modeled thus enabling auto-calibration of camera pose. The leg kinematics based velocity measurement is formulated as a right-invariant observation. Nonlinear observability analysis shows that other than the rotation around the gravity vector and the absolute position, all states are observable except for some singular cases. Discrete observability analysis demonstrates that our filter is consistent with the underlying nonlinear system. An online noise parameter tuning method is developed to adapt to the highly time-varying camera measurement noise. The proposed method is experimentally validated on a Cassie bipedal robot walking over slippery terrain. A video for the experiment can be found at https://youtu.be/VIqJL0cUr7s.

I. INTRODUCTION

For legged robots to navigate complex environments with slippery and unstable terrain illuminated with poor light, state estimation becomes important. To enable navigation in complex environments, state estimators for legged robots that fuse measurements from a wide range of sensors, such as inertial, contact and visual information are needed.

Typically, filters such as Unscented Kalman Filter (UKF), Extended Kalman Filter (EKF) and Invariant Extended Kalman Filter (InEKF) are used to fuse inertial measurements with leg kinematics for legged robot state estimation [3], [5], [18], [10]. These methods exhibit good performance when the contact point is static, as assumed in their system models. However, the drawback of these results is the degraded estimates when the contact point slips due to either slippery ground or unstable terrain. Potential slippage in these early research is either treated as noise, see [5], [10], [12], or the measurement with slippage is considered as outliers that are rejected in the update, see [3]. If the slippage is consistent or has a large magnitude, these methods can fail. Visual-inertial based methods have also been explored for state estimation. Research in [6] uses optical flow measurement for UKF innovation, where the FAST corner detector, [17], serves as the underlying algorithm to extract features. Research in [4] directly integrates the landmarks to the state space. However, fast lighting changes or loss of texture can
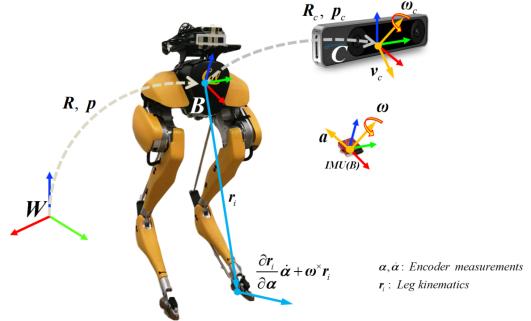
Fig. 1: The Cassie bipedal robot used for the experiment. Orientation $\boldsymbol{R}$ and position $\boldsymbol{p}$ of robot pelvis (IMU) are represented with respect to the world frame ($\boldsymbol{W}$). The tracking camera pose, i.e. $\boldsymbol{R}_c$ and $\boldsymbol{p}_c$ are expressed with respect to the robot-frame ($\boldsymbol{B}$). The tracking camera mounted on the robot's pelvis can provide velocity $\boldsymbol{v}_c$ and rotational rate $\boldsymbol{\omega}_c$ measurements in camera pose frame ($\boldsymbol{C}$).

greatly influence the quality of landmark extraction, resulting in bad state estimation.

Research in [11] and [21] use a factor graph based method that integrates inertial, leg kinematic and visual information for long range navigation in legged robots. Both researches reveal that inertial-kinematic-visual methods outperform the methods with less information fused. However, since the factor graph based methods use measurements along the whole trajectories for smoothing, the update-rate is too low for real-time control purposes.

To address slippery environments, [16] and [19] add force sensors or accelerometers, respectively, on the stance leg to detect slip events. However, these sensors are vulnerable to ground impact. A probabilistic method is used in [15] for slippage and contact detection. Contact state obtained by [15] is fused to the filter in [3] for outlier rejection. A Constraint Kalman Filter is adopted in [20] to include the linear complementary condition involved in contact modes. The external forces, friction coefficient and contact states are determined by nonlinear optimization techniques. However, information of the terrain is required to parameterize the external force, e.g. the slope of the ground, which may restrict the use of the method.

In this paper, we develop a filter-based approach that addresses the state estimation for legged robot walking over slippery terrain in a computationally efficient manner. The key idea is to fuse the vision-based velocity measurement with inertial and leg kinematics information. In particular, we use the Intel Realsense T265 tracking camera that can provide velocity and rotational rate measurements in the camera frame. The contributions of this paper are:

- Derivation of an InEKF for the inertial-legged-robot-

camera system. Measurements model for legged kinematics and tracking camera are presented.

- Auto calibration of camera pose, i.e, estimating the offset between the camera-frame and the robot-frame as part of the state estimation.
- Online noise parameter tuning to adapt to the time-varying manner of camera noise due to ground impact.
- Nonlinear and discrete observability analysis. The analysis demonstrates the consistency of our observer.
- Experimental validation using a 3D bipedal robot over slippery terrain. Experiment suggests that our method is robust to significant amount of slippage.

The proposed method builds off the inertial-kinematics based work in [3] and [10]. However, in comparison to these priori work, our method is more robust to consistent slippage, while [10] could diverge and [3] fails to reject all the slippage outliers thus leading to inaccurate estimation.

The remainder of this paper is organized as follows. Section II presents system and measurement models. Section III presents the setup of our filter. Observability analysis is presented in Section IV. Section V presents experimental result and conclusions are presented in Section VI.

II. SYSTEM AND MEASUREMENTS MODEL

For a legged robot, we wish to estimate the orientation $\boldsymbol{R} \in SO(3)$, velocity $\boldsymbol{v} \in \mathbb{R}^3$ and position $\boldsymbol{p} \in \mathbb{R}^3$ of the robot pelvis in the world frame via the measurements through the inertial measurement unit (IMU), leg kinematics and a tracking camera. The IMU is installed in the robot pelvis and we assume the IMU frame is aligned with the robot pelvis frame. In this section we introduce the model of the robot-camera system and the derive the measurement model. The robot system and the variables are illustrated in Fig. 1.

A. System Model

The IMU measures the acceleration $\boldsymbol{a}$ and angular velocity $\boldsymbol{\omega}$ in robot (IMU) frame. The measurements $\tilde{\boldsymbol{a}}$ and $\tilde{\boldsymbol{\omega}}$ are corrupted by white Gaussian noise $\boldsymbol{w}_a$, $\boldsymbol{w}_\omega$ and bias $\boldsymbol{b}_a$, $\boldsymbol{b}_\omega \in \mathbb{R}^3$:

$$\tilde{\boldsymbol{\omega}} = \boldsymbol{\omega} + \boldsymbol{b}_\omega + \boldsymbol{w}_\omega, \qquad \tilde{\boldsymbol{a}} = \boldsymbol{a} + \boldsymbol{b}_a + \boldsymbol{w}_a. \tag{1}$$

The IMU bias are modeled as random walk, i.e, their derivatives are white Gaussian noise $\boldsymbol{w}_{ba}$ and $\boldsymbol{w}_{b\omega}$, i.e.

$$\dot{\boldsymbol{b}}_\omega = \boldsymbol{w}_{b\omega}, \qquad \dot{\boldsymbol{b}}_a = \boldsymbol{w}_{ba}. \tag{2}$$

Therefore, the dynamics of robot IMU are governed by:

$$\dot{\boldsymbol{R}} = \boldsymbol{R} (\tilde{\boldsymbol{\omega}} - \boldsymbol{b}_\omega - \boldsymbol{w}_\omega)^\times, \tag{3}$$

$$\dot{\boldsymbol{p}} = \boldsymbol{v}, \qquad \dot{\boldsymbol{v}} = \boldsymbol{R} (\tilde{\boldsymbol{a}} - \boldsymbol{b}_a - \boldsymbol{w}_a) + \boldsymbol{g}. \tag{4}$$

where $\boldsymbol{g}$ is the acceleration due to gravity and $(\cdot)^\times$ denote a 3×3 skew symmetric matrix, s.t $x \times y = x^\times y, \forall x, y \in \mathbb{R}^3$.

The position and orientation of the tracking camera with respect to the robot-frame could be represented by $\boldsymbol{p}_c \in \mathbb{R}^3$ and $\boldsymbol{R}_c \in SO(3)$ respectively. We model the camera pose as constants corrupted by white Gaussian noise $\boldsymbol{w}_{pc}$ and $\boldsymbol{w}_{Rc}$:

$$\dot{\boldsymbol{R}}_c = \boldsymbol{R}_c \boldsymbol{w}_{Rc}^\times, \qquad \dot{\boldsymbol{p}}_c = \boldsymbol{w}_{pc}. \tag{5}$$

B. Leg Kinematics Measurements

The legged robot is equipped with encoders in leg joints that provide us with the corresponding angular position $\boldsymbol{\alpha}$ and its derivative $\dot{\boldsymbol{\alpha}} := \frac{\partial \boldsymbol{\alpha}}{\partial t}$. The corresponding measurements $\tilde{\boldsymbol{\alpha}}$ and $\dot{\tilde{\boldsymbol{\alpha}}}$ are corrupted by white Gaussian noise. Using leg kinematics, we can compute the location of the i_{th} contact point $\boldsymbol{r}_i(\boldsymbol{\alpha})$ in root frame. Let the position of the i_{th} contact point in world frame be $\boldsymbol{d}_i$, expressed as:

$$\boldsymbol{d}_i = \boldsymbol{p} + \boldsymbol{R}\boldsymbol{r}_i(\boldsymbol{\alpha}). \tag{6}$$

Taking derivative on both sides, we obtain the velocity of the i_{th} contact point:

$$\dot{\boldsymbol{d}}_i = \boldsymbol{v} + \boldsymbol{R}\boldsymbol{J}_i\dot{\boldsymbol{\alpha}} + \boldsymbol{R}\boldsymbol{\omega}^\times \boldsymbol{r}_i(\boldsymbol{\alpha}), \tag{7}$$

where $\boldsymbol{J}_i = \frac{\partial \boldsymbol{r}_i}{\partial \boldsymbol{\alpha}}$. Representing (7) in robot-frame, we have:

$$\boldsymbol{R}^\mathsf{T}\dot{\boldsymbol{d}}_i = \boldsymbol{R}^\mathsf{T}\boldsymbol{v} + \boldsymbol{J}_i\dot{\boldsymbol{\alpha}} + \boldsymbol{\omega}^\times \boldsymbol{r}_i(\boldsymbol{\alpha}). \tag{8}$$

If we assume the i_{th} contact point remains static to the world frame (no slip), we obtain the measurement model:

$$\boldsymbol{R}^\mathsf{T}\dot{\boldsymbol{d}}_i = 0, \qquad -\boldsymbol{J}_i\dot{\tilde{\boldsymbol{\alpha}}} - \tilde{\boldsymbol{\omega}}^\times \boldsymbol{r}_i(\tilde{\boldsymbol{\alpha}}) = \boldsymbol{R}^\mathsf{T}\boldsymbol{v} + \boldsymbol{n}_f, \tag{9}$$

where $\boldsymbol{n}_f$ is white Gaussian noise. For simplification, the single parameter $\boldsymbol{n}_f$ incorporates the uncertainty in encoder measurements, kinematic model, the effect of slip and gyroscope bias. Similar measurement model, with the gyroscope bias incorporated, is used in [3] for UKF. However, we will later show that our model could formulate a right invariant observation [2].

C. Tracking Camera Measurements

The tracking camera mounted on the top of robot could provide the measurements of velocity $\boldsymbol{v}_c$ and rotational rate $\boldsymbol{\omega}_c$ in the camera-frame. The velocity of the robot represented in the camera-frame is:

$$\boldsymbol{v}_c = \boldsymbol{R}_c^\mathsf{T}\boldsymbol{R}^\mathsf{T}\boldsymbol{v} + \boldsymbol{\omega}_c^\times \boldsymbol{R}_c^\mathsf{T}\boldsymbol{p}_c. \tag{10}$$

We assume the biases of angular velocities and rotational rate measurement have been eliminated by the tracking camera's internal visual inertial odometry algorithm. Therefore, the tracking camera measurements are modeled as solely corrupted by white Gaussian noise $\boldsymbol{n}_{vc}$ and $\boldsymbol{n}_{\omega c}$:

$$\tilde{\boldsymbol{v}}_c = \boldsymbol{v}_c + \boldsymbol{n}_{vc}, \qquad \tilde{\boldsymbol{\omega}}_c = \boldsymbol{\omega}_c + \boldsymbol{n}_{\omega c}. \tag{11}$$

The measurements finally becomes:

$$\tilde{\boldsymbol{v}}_c = \boldsymbol{R}_c^\mathsf{T}\boldsymbol{R}^\mathsf{T}\boldsymbol{v} + (\tilde{\boldsymbol{\omega}}_c - \boldsymbol{n}_{\omega c})^\times \boldsymbol{R}_c^\mathsf{T}\boldsymbol{p}_c + \boldsymbol{n}_{vc}. \tag{12}$$

D. Problem Formulation

Based on the system model (2-5) and measurement model (9, 12), we now formulate the filtering problem. For a legged robot, with one tracking camera mounted on the pelvis and leg kinematics constraints, we define the state variables:

$$\boldsymbol{x} := (\boldsymbol{R}, \ \boldsymbol{v}, \ \boldsymbol{p}, \ \boldsymbol{b}_\omega, \ \boldsymbol{b}_a, \ \boldsymbol{R}_c, \ \boldsymbol{p}_c). \tag{13}$$

Our goal is to use the measurements from the IMU, leg kinematics and tracking camera to estimate $\boldsymbol{x}$. Note, by including $\boldsymbol{R}_c$ and $\boldsymbol{p}_c$ into the state space, we will avoid the problem requiring an accurate calibration of camera pose and thus increase the robustness.

III. INVARIANT EXTENDED KALMAN FILTER SETUP

The InEKF exploits the symmetry of a system represented by a matrix Lie group. We first provide the prerequisite knowledge of InEKF and then apply it to our system. The InEKF theory and notation are presented in [2], [10], while

the InEKF augmentation of IMU bias are in [1], [10], and the system discretization is in [10].

A. Math Prerequisite

Consider a $n \times n$ matrix Lie group $\mathscr{G}$ and its associated Lie algebra $\mathfrak{g}$, representing the tangent space of $\mathscr{G}$ at the identity. We define the linear map:

$$(\cdot)^{\wedge} : \mathbb{R}^n \to \mathfrak{g}, \tag{14}$$

such that $\forall \boldsymbol{\xi} \in \mathbb{R}^n$, we can associate it with the elements in the matrix Lie group through the exponential map:

$$\exp(\cdot) : \mathbb{R}^n \to \mathscr{G}, \quad \exp(\boldsymbol{\xi}) = \exp_{\mathrm{m}}(\boldsymbol{\xi}^{\wedge}), \tag{15}$$

where $\exp_{\mathrm{m}}(\cdot)$ is the matrix exponential.

For conversion from local to global coordinate, we define the adjoint map $\mathrm{Ad}_{\boldsymbol{\chi}} : \mathfrak{g} \to \mathfrak{g}$:

$$\mathrm{Ad}_{\boldsymbol{\chi}}(\boldsymbol{\xi}^{\wedge}) = \boldsymbol{\chi} \boldsymbol{\xi}^{\wedge} \boldsymbol{\chi}^{-1}, \quad \forall \boldsymbol{\chi} \in \mathscr{G}. \tag{16}$$

The introduction of the invariant error is the core of InEKF theory. Consider a system defined on matrix Lie group $\mathscr{G}$ and its associate Lie algebra $\mathfrak{g}$ with input u_t:

$$\frac{\mathrm{d}}{\mathrm{d}t} \boldsymbol{\chi}_t = f_{u_t}(\boldsymbol{\chi}_t), \qquad \boldsymbol{\chi}_t \in \mathscr{G} \quad t \geq 0. \tag{17}$$

Consider trajectories $\boldsymbol{\chi}_t$ and $\bar{\boldsymbol{\chi}}_t$, the right invariant error between the two trajectories are defined as:

$$\boldsymbol{\eta}_t = \bar{\boldsymbol{\chi}}_t \boldsymbol{\chi}_t^{-1}. \tag{18}$$

Theorem 1 of [2] suggests that if the system in (17) is group affine, i.e:

$$f_{u_t}(\boldsymbol{\chi}_1 \boldsymbol{\chi}_2) = f_{u_t}(\boldsymbol{\chi}_1) \boldsymbol{\chi}_2 + \boldsymbol{\chi}_1 f_{u_t}(\boldsymbol{\chi}_2) - \boldsymbol{\chi}_1 f_{u_t}(\boldsymbol{I}_d) \boldsymbol{\chi}_2, \tag{19}$$

then the error dynamics are independent of state and satisfy:

$$\frac{\mathrm{d}}{\mathrm{d}t} \boldsymbol{\eta}_t = g_{u_t}(\boldsymbol{\eta}_t) := f_{u_t}(\boldsymbol{\eta}_t) - \boldsymbol{\eta}_t f_{u_t}(\boldsymbol{I}_d). \tag{20}$$

Linearizing (20) using first order approximation of the exponential map, we have:

$$\boldsymbol{\eta}_t = \exp(\boldsymbol{\xi}_t) \approx \boldsymbol{I} + \boldsymbol{\xi}_t^{\wedge}, \tag{21}$$

$$g_{u_t}(\exp(\boldsymbol{\xi}_t)) = (\boldsymbol{A}_t \boldsymbol{\xi}_t)^{\wedge} + o(\|\boldsymbol{\xi}_t\|) \approx (\boldsymbol{A}_t \boldsymbol{\xi}_t)^{\wedge}, \tag{22}$$

where the the Jacobian matrix $\boldsymbol{A}_t$ is independent of $\boldsymbol{\xi}_t$. Combining (20-22), we obtain a linear differential equation:

$$\frac{\mathrm{d}}{\mathrm{d}t} \boldsymbol{\xi}_t = \boldsymbol{A}_t \boldsymbol{\xi}_t. \tag{23}$$

The log-linear property of error propagation [2] suggests that if the initial error satisfies $\boldsymbol{\eta}_0 = \exp(\boldsymbol{\xi}_0)$, the linearized error dynamics (23) can fully represent the nonlinear error defined in (18), as:

$$\boldsymbol{\eta}_t = \exp(\boldsymbol{\xi}_t), \quad t \geq 0. \tag{24}$$

In general, for systems satisfying the group affine property, by the introduction of the invariant error and the log-linear property of error propagation, we can use a linear differential equation (23) to describe the error between two states thus avoiding problems involved in nonlinear observer design.

B. Estimation Error Dynamics

Based on the system model (2-5), we give the definition of the error between the estimated states $\hat{\boldsymbol{x}}$ and the real states $\boldsymbol{x}$, where $\hat{(\cdot)}$ denotes the estimated states.

The states of IMU $(\boldsymbol{R}, \boldsymbol{v}, \boldsymbol{p})$ are embedded in the double direct spatial isometry $SE_2(3)$ [2]:

$$\boldsymbol{\chi} := \begin{bmatrix} \boldsymbol{R} & \boldsymbol{v} & \boldsymbol{p} \\ \boldsymbol{0}_{1\times3} & 1 & 0 \\ \boldsymbol{0}_{1\times3} & 0 & 1 \end{bmatrix}. \tag{25}$$

This matrix is the extension of special Euclidean group $SE(3)$ with additional spatial vectors. Without bias, the

dynamics of IMU states are group affine. Thus, we define the right-invariant error of the IMU states and linearize it:

$$\boldsymbol{\eta} = \hat{\boldsymbol{\chi}} \boldsymbol{\chi}^{-1}$$
$$= \begin{bmatrix} \hat{\boldsymbol{R}} \boldsymbol{R}^{\mathsf{T}} & \hat{\boldsymbol{v}} - \hat{\boldsymbol{R}} \boldsymbol{R}^{\mathsf{T}} \boldsymbol{v} & \hat{\boldsymbol{p}} - \hat{\boldsymbol{R}} \boldsymbol{R}^{\mathsf{T}} \boldsymbol{p} \\ \boldsymbol{0}_{1\times3} & 1 & 0 \\ \boldsymbol{0}_{1\times3} & 0 & 1 \end{bmatrix} \approx \begin{bmatrix} \boldsymbol{I} + \boldsymbol{\xi}_R^{\wedge} & \boldsymbol{\xi}_v & \boldsymbol{\xi}_p \\ \boldsymbol{0}_{1\times3} & 1 & 0 \\ \boldsymbol{0}_{1\times3} & 0 & 1 \end{bmatrix}, \tag{26}$$

We define $\boldsymbol{\xi}_{IMU} := [\boldsymbol{\xi}_R^{\mathsf{T}}, \boldsymbol{\xi}_v^{\mathsf{T}}, \boldsymbol{\xi}_p^{\mathsf{T}}]^{\mathsf{T}} \in \mathbb{R}^9$. Using $\boldsymbol{\xi}_{IMU}$ we can formulate the standard InEKF. However, augmenting the error of IMU biases and camera misalignment will result in an "imperfect" InEKF [1], i.e, the state matrix of linearized system (23) becomes state dependent.

The estimation error of $\boldsymbol{R}_c$ is defined by:

$$\eta_c = \hat{\boldsymbol{R}}_c \boldsymbol{R}_c^{\mathsf{T}} = \exp(\boldsymbol{\xi}_{R_c}) \approx \boldsymbol{I} + \boldsymbol{\xi}_{R_c}^{\wedge}. \tag{27}$$

For IMU biases and camera position, the estimation error are calculated by standard vector difference, i.e:

$$\boldsymbol{e}_{b\omega} = \hat{\boldsymbol{b}}_{\boldsymbol{\omega}} - \boldsymbol{b}_{\boldsymbol{\omega}}, \quad \boldsymbol{e}_{ba} = \hat{\boldsymbol{b}}_{\boldsymbol{a}} - \boldsymbol{b}_{\boldsymbol{a}}, \quad \boldsymbol{e}_{pc} = \hat{\boldsymbol{p}}_c - \boldsymbol{p}_c. \tag{28}$$

The linearized estimation error of the entire system could be expressed as the concatenation of the error in (26-28):

$$\boldsymbol{\xi} := [\boldsymbol{\xi}_{IMU}^{\mathsf{T}}, \boldsymbol{e}_{b\omega}^{\mathsf{T}}, \boldsymbol{e}_{ba}^{\mathsf{T}}, \boldsymbol{\xi}_{R_c}^{\mathsf{T}}, \boldsymbol{e}_{pc}^{\mathsf{T}}]^{\mathsf{T}} \in \mathbb{R}^{21}. \tag{29}$$

By system dynamics (2-5) and equation (20-22), we could derive the linearized estimation error dynamics:

$$\frac{d}{dt} \left(\hat{\boldsymbol{R}} \boldsymbol{R}^{\mathsf{T}} \right) \approx \left(\hat{\boldsymbol{R}} (\boldsymbol{w}_{\omega} - \boldsymbol{e}_{b\omega}) \right)^{\times},$$

$$\frac{d}{dt} \left(\hat{\boldsymbol{v}} - \hat{\boldsymbol{R}} \boldsymbol{R}^{\mathsf{T}} \boldsymbol{v} \right) \approx \boldsymbol{g}^{\times} \boldsymbol{\xi}_R + \hat{\boldsymbol{R}} (\boldsymbol{w}_a - \boldsymbol{e}_{ba})$$
$$+ \hat{\boldsymbol{v}}^{\times} \boldsymbol{\xi}_R (\boldsymbol{w}_{\omega} - \boldsymbol{e}_{b\omega}),$$

$$\frac{d}{dt} \left(\hat{\boldsymbol{p}} - \hat{\boldsymbol{R}} \boldsymbol{R}^{\mathsf{T}} \boldsymbol{p} \right) \approx \boldsymbol{\xi}_v + \hat{\boldsymbol{p}}^{\times} \hat{\boldsymbol{R}} (\boldsymbol{w}_{\omega} - \boldsymbol{e}_{b\omega}) \tag{30}$$

$$\frac{d}{dt} \boldsymbol{e}_{b\omega} = \boldsymbol{w}_{b\omega}, \qquad \frac{d}{dt} \boldsymbol{e}_{b\omega} = \boldsymbol{w}_{ba},$$

$$\frac{d}{dt} \left(\hat{\boldsymbol{R}}_c \boldsymbol{R}_c^{\mathsf{T}} \right) = \boldsymbol{w}_{Rc}^{\times}, \qquad \frac{d}{dt} \boldsymbol{e}_{pc} = \boldsymbol{w}_{pc}.$$

Representing (30) in matrix form, we have:

$$\dot{\boldsymbol{\xi}} = \boldsymbol{A} \boldsymbol{\xi} + \boldsymbol{B}_{\hat{\boldsymbol{x}}} \boldsymbol{w}, \tag{31}$$

where,

$$\boldsymbol{A} = \begin{bmatrix} \boldsymbol{0}_3 & \boldsymbol{0}_3 & \boldsymbol{0}_3 & -\hat{\boldsymbol{R}} & \boldsymbol{0}_3 & \boldsymbol{0}_{3\times6} \\ \boldsymbol{g}^{\times} & \boldsymbol{0}_3 & \boldsymbol{0}_3 & -\hat{\boldsymbol{v}}^{\times} \hat{\boldsymbol{R}} & -\hat{\boldsymbol{R}} & \boldsymbol{0}_{3\times6} \\ \boldsymbol{0}_3 & \boldsymbol{I}_3 & \boldsymbol{0}_3 & -\hat{\boldsymbol{p}}^{\times} \hat{\boldsymbol{R}} & \boldsymbol{0}_3 & \boldsymbol{0}_{3\times6} \\ & & \boldsymbol{0}_{12\times9} & & & \boldsymbol{I}_{12} \end{bmatrix}, \tag{32}$$

$$\boldsymbol{B}_{\hat{\boldsymbol{x}}} = \begin{bmatrix} \mathrm{Ad}_{\hat{\boldsymbol{\chi}}} & \boldsymbol{0}_{9\times12} \\ \boldsymbol{0}_{12\times9} & \boldsymbol{I}_{12} \end{bmatrix}, \mathrm{Ad}_{\hat{\boldsymbol{\chi}}} = \begin{bmatrix} \hat{\boldsymbol{R}} & \boldsymbol{0}_3 & \boldsymbol{0}_3 \\ -\hat{\boldsymbol{v}}^{\times} \hat{\boldsymbol{R}} & \hat{\boldsymbol{R}} & \boldsymbol{0}_3 \\ -\hat{\boldsymbol{p}}^{\times} \hat{\boldsymbol{R}} & \boldsymbol{0}_3 & \hat{\boldsymbol{R}} \end{bmatrix}, \tag{33}$$

$$\boldsymbol{w} := \mathrm{vec}(\boldsymbol{w}_{\omega}, \boldsymbol{w}_a, \boldsymbol{0}_{3\times1}, \boldsymbol{w}_{ba}, \boldsymbol{w}_{b\omega}, \boldsymbol{w}_{R_c}, \boldsymbol{w}_{p_c}). \tag{34}$$

Note $\boldsymbol{w}$ corresponds to the noise and the system covariance is $\boldsymbol{P}$. The dynamics of $\boldsymbol{P}$ is governed by the Ricatti equation given as follows:

$$\frac{\mathrm{d}}{\mathrm{dt}} \boldsymbol{P} = \boldsymbol{A} \boldsymbol{P} + \boldsymbol{P} \boldsymbol{A}^{\mathsf{T}} + \boldsymbol{Q}, \quad \boldsymbol{Q} = \boldsymbol{B}_{\hat{\boldsymbol{x}}} \mathrm{Cov}(\boldsymbol{w}) \boldsymbol{B}_{\hat{\boldsymbol{x}}}^{\mathsf{T}}. \tag{35}$$

C. Propagation Steps

To apply the filter in discrete time, we assume zero-order hold to the input and perform Euler integration from time t_k to t_{k+1}. The discrete dynamics becomes:

$$\hat{\boldsymbol{R}}_{k+1}^{-} = \hat{\boldsymbol{R}}_k^{+} \exp \left(\left(\tilde{\boldsymbol{\omega}}_k - \hat{\boldsymbol{b}}_{\omega,k}^{+} \right) \Delta t \right), \tag{36}$$

$$\hat{\boldsymbol{v}}_{k+1}^{-} = \hat{\boldsymbol{v}}_k^{+} + \hat{\boldsymbol{R}}_k^{+} \left(\tilde{\boldsymbol{a}}_k - \hat{\boldsymbol{b}}_{a,k}^{+} \right) \Delta t + \boldsymbol{g} \Delta t, \tag{37}$$

$$\hat{\boldsymbol{p}}_{k+1}^{-} = \hat{\boldsymbol{p}}_k^{+} + \hat{\boldsymbol{v}}_k^{+} \Delta t + \frac{1}{2} \hat{\boldsymbol{R}}_k^{+} \left(\tilde{\boldsymbol{a}}_k - \hat{\boldsymbol{b}}_{a,k}^{+} \right) \Delta t^2 + \frac{1}{2} \boldsymbol{g} \Delta t^2, \tag{38}$$

$$\hat{\boldsymbol{d}}_{k+1}^{-} = \hat{\boldsymbol{d}}_k^{+}, \quad \hat{\boldsymbol{b}}_{\omega,k+1}^{-} = \hat{\boldsymbol{b}}_{\omega,k}^{+}, \quad \hat{\boldsymbol{b}}_{a,k+1}^{-} = \hat{\boldsymbol{b}}_{a,k}^{+}, \tag{39}$$

$$\hat{\boldsymbol{R}}_{c,k+1}^{-} = \hat{\boldsymbol{R}}_{c,k}^{+}, \qquad \hat{\boldsymbol{p}}_{c,k+1}^{-} = \hat{\boldsymbol{p}}_{c,k}^{+}. \tag{40}$$

where $\Delta t = t_{k+1} - t_k, (\cdot)_k^+$ denotes the estimated states at time t_k with all measurements until t_k are processed and $(\cdot)_k^-$ denotes the state estimated at time t_k through propagation. Given the system matrix (32) at time t_k, we have discrete state transformation matrix and discrete-time covariance propagation equation:

$$\Phi_k = \exp_{\mathrm{m}}(\boldsymbol{A}_k \Delta t), \quad \boldsymbol{P}_{k+1} = \Phi_k \boldsymbol{P}_k \Phi_k^{\mathsf{T}} + \boldsymbol{Q}_k, \tag{41}$$

where $\boldsymbol{Q}_k \approx \Phi_k \boldsymbol{Q} \Phi_k^{\mathsf{T}} \Delta t$. We recommend [10] for more details on the system discretization.

D. Update Steps

The leg kinematics measurement (9) formulates the right-invariant observation [2], which exploits the geometry property of the system:

$$\boldsymbol{y} = \boldsymbol{\chi}^{-1} \boldsymbol{b} + \boldsymbol{s}, \tag{42}$$

where:

$$\boldsymbol{y} = [-(\boldsymbol{J}_i \dot{\tilde{\boldsymbol{\alpha}}} + \tilde{\boldsymbol{\omega}}^{\times} \boldsymbol{r}_i(\tilde{\boldsymbol{\alpha}}))^{\mathsf{T}}, -1, 0]^{\mathsf{T}}, \tag{43}$$

$$\boldsymbol{b} = [\boldsymbol{0}_{1\times 3}, -1, 0]^{\mathsf{T}}, \quad \boldsymbol{s} = \left[\boldsymbol{n}_f^{\mathsf{T}}, 0, 0\right]^{\mathsf{T}}. \tag{44}$$

As the right-invariant observation only contains elements in $\boldsymbol{\chi}$, we first derive the innovation term for $\boldsymbol{\chi}$ and then expand it to the full state. The update equation for $\boldsymbol{\chi}$ are defined by matrix multiplication:

$$\hat{\boldsymbol{\chi}}^{+} = \exp\left(\boldsymbol{L}\left(\hat{\boldsymbol{\chi}}^{-} \boldsymbol{y} - \boldsymbol{b}\right)\right) \hat{\boldsymbol{\chi}}^{-}, \tag{45}$$

$$\boldsymbol{\eta}^{+} = \exp\left(\boldsymbol{L}\left(\boldsymbol{\eta}^{-} \boldsymbol{b} - \boldsymbol{b} + \boldsymbol{\chi}^{-} \boldsymbol{s}\right)\right) \hat{\boldsymbol{\eta}}^{-}, \tag{46}$$

where $\boldsymbol{L}$ is the observation gain matrix. Using the first order approximation of exponential map, we have:

$$\boldsymbol{\eta}^{+} = \exp(\boldsymbol{\xi}_{IMU}^{+}) \approx (\boldsymbol{I} + \boldsymbol{\xi}_{IMU}^{+})^{\wedge} \\ \approx (\boldsymbol{I} + \boldsymbol{\xi}_{IMU}^{+})^{\wedge} + \left(\boldsymbol{L}\boldsymbol{\Pi}\left(\boldsymbol{\eta}^{-} \boldsymbol{b} - \boldsymbol{b} + \boldsymbol{\chi}^{-} \boldsymbol{s}\right)\right)^{\wedge}, \tag{47}$$

where $\boldsymbol{\Pi} = [\boldsymbol{I}_3, \boldsymbol{0}_{3\times 2}]$ is an auxiliary matrix to select the first three rows of the right hand side. Therefore we have the update equation for $\boldsymbol{\xi}_{IMU}$:

$$\boldsymbol{\xi}_{IMU}^{+} = \boldsymbol{\xi}_{IMU}^{-} - \boldsymbol{L}\left([\boldsymbol{0}_3, -\boldsymbol{I}_3, \boldsymbol{0}_3]\boldsymbol{\xi}_{IMU}^{-} - [\boldsymbol{0}_{1\times 3}, (\hat{\boldsymbol{R}}\boldsymbol{n}_f)^{\mathsf{T}}, \boldsymbol{0}_{1\times 3}]^{\mathsf{T}}\right). \tag{48}$$

Expand (48) to full state (29), we have:

$$\boldsymbol{\xi}^{+} = \boldsymbol{\xi}^{-} - \boldsymbol{K}\left(\boldsymbol{H}\boldsymbol{\xi}^{-} - [\boldsymbol{0}_{1\times 3}, (\hat{\boldsymbol{R}}\boldsymbol{n}_f)^{\mathsf{T}}, \boldsymbol{0}_{1\times 15}]^{\mathsf{T}}\right), \tag{49}$$

where the Kalman gain $\boldsymbol{K}$ is given by:

$$\boldsymbol{K} = \boldsymbol{P}\boldsymbol{H}^{\mathsf{T}}\boldsymbol{S}^{-1}, \quad \text{with } \boldsymbol{S} = \boldsymbol{H}\boldsymbol{P}^{-}\boldsymbol{H}^{\mathsf{T}} + \boldsymbol{N}, \tag{50}$$

$$\boldsymbol{N} = \hat{\boldsymbol{R}}\operatorname{Cov}(\boldsymbol{n}_f)\hat{\boldsymbol{R}}^{\mathsf{T}}, \quad \boldsymbol{H} = [\boldsymbol{0}_3, -\boldsymbol{I}_3, \boldsymbol{0}_{3\times 15}]. \tag{51}$$

Using the correction term:

$$\boldsymbol{\delta} := \left[\boldsymbol{\delta}_{IMU}^{\mathsf{T}}, \boldsymbol{\delta}_{b\omega}^{\mathsf{T}}, \boldsymbol{\delta}_{ba}^{\mathsf{T}}, \boldsymbol{\delta}_{Rc}^{\mathsf{T}}, \boldsymbol{\delta}_{pc}^{\mathsf{T}}\right]^{\mathsf{T}} = \boldsymbol{K}\boldsymbol{\Pi}(\hat{\boldsymbol{\chi}}\boldsymbol{y} - \boldsymbol{b}). \tag{52}$$

we finally have the update equations:

$$\hat{\boldsymbol{\chi}}^{+} = \exp(\boldsymbol{\delta}_{IMU})\hat{\boldsymbol{\chi}}^{-}, \tag{53}$$

$$\boldsymbol{b}_a^{+} = \boldsymbol{b}_a^{-} + \boldsymbol{\delta}_{ba}, \quad \boldsymbol{b}_\omega^{+} = \boldsymbol{b}_\omega^{-} + \boldsymbol{\delta}_{b\omega}, \tag{54}$$

$$\boldsymbol{R}_c^{+} = \exp(\boldsymbol{\delta}_{Rc})\boldsymbol{R}_c^{-}, \quad \boldsymbol{p}_c^{+} = \boldsymbol{p}_c^{-} + \boldsymbol{\delta}_{pc}, \tag{55}$$

$$\boldsymbol{P}^{+} = (\boldsymbol{I} - \boldsymbol{K}\boldsymbol{H})\boldsymbol{P}^{-}(\boldsymbol{I} - \boldsymbol{K}\boldsymbol{H})^{\mathsf{T}} + \boldsymbol{K}\boldsymbol{N}\boldsymbol{K}^{\mathsf{T}}. \tag{56}$$

The tracking camera measurements (12) can not be formulated as a right invariant observation. Therefore, the correction term $\boldsymbol{\delta}$ will be computed by standard vector addition. A pseudo measurement based InEKF implementation using a similar update equation has been reported in [7]. For measurement $\boldsymbol{y} = \boldsymbol{h}(\boldsymbol{x}, \boldsymbol{n})$, we implement first order approximation to obtain the linearized observation model:

$$\boldsymbol{h}(\hat{\boldsymbol{x}}, \boldsymbol{0}) - \boldsymbol{h}(\boldsymbol{x}, \boldsymbol{n}) = \boldsymbol{H}_c \boldsymbol{\xi} + \boldsymbol{G}\boldsymbol{n} + h.o.t(\boldsymbol{\xi}, \boldsymbol{n}), \tag{57}$$

where $\boldsymbol{H}_c$ and $\boldsymbol{G}$ are Jacobians with respect to the state error and noise term respectively.

We apply (57) to (12) to compute the the observation matrix for the tracking camera measurements:

$$\boldsymbol{H}_c = \left[\boldsymbol{0}, \hat{\boldsymbol{R}}_c^{\mathsf{T}}\hat{\boldsymbol{R}}^{\mathsf{T}}, \boldsymbol{0}_{3\times 9}, \boldsymbol{\omega}_c^{\times}\hat{\boldsymbol{R}}_c^{\mathsf{T}}\hat{\boldsymbol{p}}_c^{\times} + \hat{\boldsymbol{R}}_c^{\mathsf{T}}(\hat{\boldsymbol{R}}^{\mathsf{T}}\hat{\boldsymbol{v}})^{\times}, -\boldsymbol{\omega}_c^{\times}\hat{\boldsymbol{R}}^{\mathsf{T}}\right]. \tag{58}$$

Note that we need to take the derivative w.r.t to error defined in (29) to obtain (58). The corresponding covariance matrix for this measurements takes the form:

$$\boldsymbol{N}_c = \hat{\boldsymbol{R}}\hat{\boldsymbol{R}}_c \operatorname{Cov}(\tilde{\boldsymbol{v}}_c)\hat{\boldsymbol{R}}_c^{\mathsf{T}}\hat{\boldsymbol{R}}^{\mathsf{T}}, \tag{59}$$

$$\operatorname{Cov}(\tilde{\boldsymbol{v}}_c) = \operatorname{Cov}(\boldsymbol{n}_{vc}) + (-\hat{\boldsymbol{R}}_c^{\mathsf{T}}\hat{\boldsymbol{p}}_c)^{\times}\operatorname{Cov}(\boldsymbol{n}_{\omega c})(\hat{\boldsymbol{R}}_c^{\mathsf{T}}\hat{\boldsymbol{p}}_c)^{\times}. \tag{60}$$

Replacing $\boldsymbol{N}$ and $\boldsymbol{H}$ in equations (50-51) with $\boldsymbol{N}_c$ and $\boldsymbol{H}_c$, we have the update term for the tracking camera measurements:

$$\boldsymbol{\delta}_c = \boldsymbol{K}(\boldsymbol{h}(\hat{\boldsymbol{x}}) - \tilde{\boldsymbol{v}}_c). \tag{61}$$

It is worth noticing that the observation matrix $\boldsymbol{H}$ (51) for leg kinematics measurement is a constant while $\boldsymbol{H}_c$ (58) is state dependent. The state-dependent nature of $\boldsymbol{H}_c$ may lead to inconsistency in observability and thus needs more attention.

E. Parameter Tuning

As an accurate estimation of sensor noise is key to the implementation of filtering, the covariance of the tracking camera measurements should be carefully computed. We define the tracking camera measurement $\tilde{\boldsymbol{m}} := [\tilde{\boldsymbol{\omega}}_c^{\mathsf{T}}, \tilde{\boldsymbol{v}}_c^{\mathsf{T}}]^{\mathsf{T}}$ and its covariance $\tilde{\boldsymbol{w}}_m := [\boldsymbol{n}_{\omega c}^{\mathsf{T}}, \boldsymbol{n}_{vc}^{\mathsf{T}}]^{\mathsf{T}}$. Since the periodic ground impact while walking causes highly time-varying camera noise, we estimate covariance of $\tilde{\boldsymbol{w}}_m$ online rather than using a fixed value. The covariance of $\tilde{\boldsymbol{w}}_m$ at time t_k is denoted by $Cov(\boldsymbol{w}_m)_k$, which is estimated by the empirical covariance from t_{k-n} to $t_k, n \in \mathbb{N}^{+}$:

$$Cov(\boldsymbol{w}_m)_k \approx \frac{1}{n}\sum_{i=0}^{n} \boldsymbol{e}_{k-i}\boldsymbol{e}_{k-i}^{\mathsf{T}}, \text{with } \boldsymbol{e}_{k-i} = \tilde{\boldsymbol{m}}_{t_{k-i}} - \frac{1}{n}\sum_{j=0}^{n} \tilde{\boldsymbol{m}}_{t_{k-j}}. \tag{62}$$

To avoid large delays, we tend to use a small n. In the experiment, the filter works well using online tuned $Cov(\boldsymbol{w}_m)_k$ but diverges when we use a fixed value. The estimated covariance is presented in Fig. 3, which is highly time-varying.

IV. OBSERVABILITY ANALYSIS

This section discusses the observability of the filter. As the leg kinematic measurements forms a right-invariant observation, we could obtain the unobservable states of the filter without nonlinear observability analysis [2]. Similar systems has been thoroughly studied in [3] and [10], thus we only analyze the filter with tracking camera measurements.

A. Nonlinear Observability Analysis

We employ the notion of locally weak observability depicted in [13]. Consider a nonlinear system in the form:

$$\dot{\boldsymbol{x}} = \boldsymbol{f}(\boldsymbol{x}, \boldsymbol{u}), \quad \boldsymbol{y} = \boldsymbol{h}(\boldsymbol{x}). \tag{63}$$

Given state $\boldsymbol{x}$ and input $\boldsymbol{u}$, the observability matrix is spanned by the gradients of the Lie derivatives:

$$\boldsymbol{O}(\boldsymbol{x}, \boldsymbol{u}) = \begin{bmatrix} \nabla \mathscr{L}_f^0 \boldsymbol{h}(\boldsymbol{x}) \\ \nabla \mathscr{L}_f^1 \boldsymbol{h}(\boldsymbol{x}) \\ \vdots \end{bmatrix}, \tag{64}$$

$$\mathscr{L}_f^0 \boldsymbol{h}(\boldsymbol{x}) = \boldsymbol{h}(\boldsymbol{x}), \dots \mathscr{L}_f^n \boldsymbol{h}(\boldsymbol{x}) = \frac{\partial \mathscr{L}_f(\boldsymbol{x})^{n-1}\boldsymbol{h}}{\partial \boldsymbol{x}}\boldsymbol{f}, \quad n \geq 1. \tag{65}$$

We use exponential coordinates to parameterize rotation:

$$\boldsymbol{R} = \exp(\boldsymbol{\phi}), \quad \dot{\boldsymbol{\phi}} = \boldsymbol{R}(\boldsymbol{\phi})\boldsymbol{\omega}, \quad \boldsymbol{\phi} := [\phi_x, \phi_y, \phi_z]^{\mathsf{T}}, \tag{66}$$

where $\boldsymbol{\omega}$ could be interpreted as the rotational rate expressed in the robot-frame while $\dot{\boldsymbol{\phi}}$ is in the world frame. Similarly, with $\boldsymbol{R}_c = \exp(\boldsymbol{\phi}_c)$, we then represent the system (2-5) in robot-centric frame to lower the computation burden:

$$\begin{aligned}\bar{\boldsymbol{x}} &:= \begin{bmatrix} \bar{\boldsymbol{\phi}}^\top, \bar{\boldsymbol{v}}^\top, \bar{\boldsymbol{p}}^\top, \bar{\boldsymbol{b}}_a^\top, \bar{\boldsymbol{b}}_\omega^\top, \bar{\boldsymbol{\phi}}_c^\top, \bar{\boldsymbol{p}}_c^\top \end{bmatrix}^\top \\ &= \begin{bmatrix} \boldsymbol{\phi}^\top, (\boldsymbol{R}^\top \boldsymbol{v})^\top, (\boldsymbol{R}^\top \boldsymbol{p})^\top, \boldsymbol{b}_a^\top, \boldsymbol{b}_\omega^\top, \boldsymbol{\phi}_c^\top, \boldsymbol{p}_c^\top \end{bmatrix}^\top.\end{aligned} \quad (67)$$

As the transformation (67) is invertible, the observability will not change. The system becomes:

$$\dot{\boldsymbol{x}} = \boldsymbol{f}(\bar{\boldsymbol{x}}, \bar{\boldsymbol{u}}) := \begin{bmatrix} \bar{\boldsymbol{R}} \bar{\boldsymbol{\omega}} \\ \bar{\boldsymbol{\omega}}^\times \bar{\boldsymbol{v}} + \bar{\boldsymbol{a}} + \bar{\boldsymbol{R}}^\top \boldsymbol{g} \\ \bar{\boldsymbol{\omega}}^\times \bar{\boldsymbol{p}} + \bar{\boldsymbol{v}} \\ \boldsymbol{0}_{12 \times 1} \end{bmatrix}, \quad (68)$$

$$\boldsymbol{h}_c(\bar{\boldsymbol{x}}) = \bar{\boldsymbol{R}}_c^\top \bar{\boldsymbol{v}} + \bar{\boldsymbol{\omega}}_c^\times \bar{\boldsymbol{R}}_c^\top \bar{\boldsymbol{p}}_c, \quad (69)$$

$$\bar{\boldsymbol{\omega}} := \boldsymbol{\omega} - \bar{\boldsymbol{b}}_\omega, \quad \bar{\boldsymbol{a}} := \boldsymbol{a} - \bar{\boldsymbol{b}}_a, \quad \bar{\boldsymbol{u}} := vec(\bar{\boldsymbol{\omega}}, \bar{\boldsymbol{a}}). \quad (70)$$

Applying (64) to (68, 69) and do row transformation, the observability matrix could be obtained and simplified as:

$$\boldsymbol{O}_c = \begin{bmatrix} \boldsymbol{0} & \boldsymbol{I} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \Delta & (\bar{\boldsymbol{R}}_c \boldsymbol{\omega}_c)^\times \\ \bar{\boldsymbol{R}}^\top \boldsymbol{g}^\times & \bar{\boldsymbol{\omega}}^\times & \boldsymbol{0} & \bar{\boldsymbol{v}}^\times & -\boldsymbol{I} & \#_{0,6} & \boldsymbol{0} \\ \boldsymbol{0} & \bar{\boldsymbol{\omega}}^{\times 2} & \boldsymbol{0} & \#_{2,4} & -\bar{\boldsymbol{\omega}}^\times & \#_{1,6} & \boldsymbol{0} \\ \#_{2,1} & \bar{\boldsymbol{\omega}}^{\times 3} & \boldsymbol{0} & \#_{3,4} & -\bar{\boldsymbol{\omega}}^{\times 2} & \#_{2,6} & \boldsymbol{0} \\ \boldsymbol{0} & \bar{\boldsymbol{\omega}}^{\times 4} & \boldsymbol{0} & \#_{4,4} & -\bar{\boldsymbol{\omega}}^{\times 3} & \#_{3,6} & \boldsymbol{0} \\ \#_{4,1} & \bar{\boldsymbol{\omega}}^{\times 5} & \boldsymbol{0} & \#_{5,4} & -\bar{\boldsymbol{\omega}}^{\times 4} & \#_{4,6} & \boldsymbol{0} \\ \boldsymbol{0} & \bar{\boldsymbol{\omega}}^{\times 6} & \boldsymbol{0} & \#_{6,4} & -\bar{\boldsymbol{\omega}}^{\times 5} & \#_{5,6} & \boldsymbol{0} \end{bmatrix}, \quad (71)$$

$$\#_{i,1} = \bar{\boldsymbol{\omega}}^{\times i} \bar{\boldsymbol{R}}^\top \boldsymbol{g}^\times, \quad \#_{i,4} = \partial \bar{\boldsymbol{\omega}}^{\times i} \bar{\boldsymbol{v}} / \partial \bar{\boldsymbol{b}}_\omega,$$

$$\Delta = \bar{\boldsymbol{v}}^\times + (\bar{\boldsymbol{R}}_c \boldsymbol{\omega}_c)^\times \boldsymbol{p}_c, \quad \#_{0,6} = (\bar{\boldsymbol{\omega}}^\times \bar{\boldsymbol{v}} + \bar{\boldsymbol{a}} + \bar{\boldsymbol{R}}^\top \boldsymbol{g}^\times)^\times, \quad (72)$$

$$\#_{i,6} = \left(\bar{\boldsymbol{\omega}}^{\times i} \left(\bar{\boldsymbol{\omega}}^\times \bar{\boldsymbol{v}} + \bar{\boldsymbol{a}} + \frac{1 + (-1)^i}{2} \bar{\boldsymbol{R}}^\top \boldsymbol{g}^\times \right) \right)^\times.$$

Without the effect of noise, we also have $\bar{\boldsymbol{\omega}}^\times = (\bar{\boldsymbol{R}}_c \boldsymbol{\omega}_c)^\times$. Therefore we could determine the null space of the observability matrix, i.e. $\bar{\boldsymbol{U}}_c$ s.t. $\bar{\boldsymbol{O}}_c \bar{\boldsymbol{U}}_c = \boldsymbol{0}$:

$$\boldsymbol{U}_c = \begin{bmatrix} \boldsymbol{g}^\top & 0 & 0 & 0 & 0 & 0 & 0 \\ \boldsymbol{0} & 0 & \boldsymbol{I} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & \bar{\boldsymbol{\omega}}^\top \end{bmatrix}^\top. \quad (73)$$

The first two rows of $\boldsymbol{U}_c^\top$ suggest that the orientation around the gravity vector and the position are unobservable. The position of the camera along the body angular velocity is also unobservable with only one measurement according to the last row. Note, the camera position will be fully observable when multiple different measurements are obtained.

The observability matrix may lose rank with certain robot motion, see the simulation of camera pose estimation in the video. We next list the singular cases based on the angular and linear velocity of robot pelvis. If $\bar{\boldsymbol{\omega}} = 0, \bar{\boldsymbol{v}} = 0$, the last 2 columns of $\boldsymbol{O}_c$ in (71) become 0 thus rank loss is 5. In this case, the camera pose becomes fully unobservable. If $\bar{\boldsymbol{\omega}} = 0, \bar{\boldsymbol{v}} \neq 0$, the last column of $\boldsymbol{O}_c$ becomes zero, while the rank of the last but one column depends on $\bar{\boldsymbol{v}}^\times$. Thus the total rank loss is 3. In this case, the orientation around the velocity vector and camera position are unobservable.

B. Discrete Observability Analysis

The filter is applied in linearized and discretized version:

$$\boldsymbol{x}_{k+1} = \boldsymbol{\Phi}_k \boldsymbol{x}_k, \quad \boldsymbol{y}_k = \boldsymbol{H}_k \boldsymbol{x}_k. \quad (74)$$

However, observability of this discrete linear time-varying system may differ from the underlying nonlinear system as the linearized points deviate from the real states. This inconsistency in EKF-SLAM has been analyzed and solved

by Observability Constraint EKF [14]. Similar problems in EKF design for legged robot navigation are also studied in [5]. Following the method in [14], we analyze the local observability matrix [8] at the "standard" operating point, i.e, linearized at the latest estimated value:

$$\hat{\boldsymbol{O}} = \begin{bmatrix} \boldsymbol{H}_k^- \\ \boldsymbol{H}_{k+1}^- \boldsymbol{\Phi}_k^+ \\ \boldsymbol{H}_{k+2}^- \boldsymbol{\Phi}_{k+1}^+ \boldsymbol{\Phi}_k^+ \\ \vdots \end{bmatrix}. \quad (75)$$

By applying (75), the discrete observability matrix could be obtained and simplified to:

$$\hat{\boldsymbol{O}}_c = \begin{bmatrix} \boldsymbol{0} & \boldsymbol{R}_k^{-\top} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \#_{0,6} & \boldsymbol{R}_{c,k}^- \boldsymbol{\omega}_{c,k}^\times \boldsymbol{R}_{c,k}^{-\top} \\ \boldsymbol{R}_{k+1}^{-\top} \boldsymbol{g}^\times \Delta t & \#_{1,2} & \boldsymbol{0} & \#_{1,4} & \#_{1,5} & \#_{1,6} & \#_{1,7} \\ \#_{2,1} & \#_{2,2} & \boldsymbol{0} & \#_{2,4} & \#_{2,5} & \#_{2,6} & \#_{2,7} \\ \#_{3,1} & \#_{3,2} & \boldsymbol{0} & \#_{3,4} & \#_{3,5} & \#_{3,6} & \#_{3,7} \\ \#_{4,1} & \#_{4,2} & \boldsymbol{0} & \#_{4,4} & \#_{4,5} & \#_{4,6} & \#_{4,7} \\ \#_{5,1} & \#_{5,2} & \boldsymbol{0} & \#_{5,4} & \#_{5,5} & \#_{5,6} & \#_{5,7} \\ \#_{6,1} & \#_{6,2} & \boldsymbol{0} & \#_{6,4} & \#_{6,5} & \#_{6,6} & \#_{6,7} \end{bmatrix} \quad (76)$$

$$\#_{i,1} = i \boldsymbol{R}_{k+i}^{-\top} \boldsymbol{g}^\times \Delta t, i \geq 1, \#_{i,2} = \boldsymbol{R}_{k+i}^{-\top}, i \geq 1$$

$$\#_{1,4} = -\boldsymbol{R}_{k+1}^{-\top} \left(\boldsymbol{v}_k^{+\times} \boldsymbol{R}_k^+ \Delta t + \frac{1}{2} \boldsymbol{g}^\times \boldsymbol{R}_k^+ \Delta t^2 \right)$$

$$\#_{i,4} = -\boldsymbol{R}_{k+1}^{-\top} \sum_{j=0}^{i-1} \left(\boldsymbol{v}_{k+j}^{+\times} \boldsymbol{R}_{k+j}^+ \Delta t + \frac{1}{2} \boldsymbol{g}^\times \boldsymbol{R}_{k+j}^+ \Delta t^2 \right) - \quad (77)$$

$$\boldsymbol{R}_{k+i}^{-\top} \sum_{l=0}^{i-2} (i - l - 1) \boldsymbol{g}^\times \boldsymbol{R}_{k+l}^+ \Delta t^2, i \geq 2$$

$$\#_{i,5} = -\sum_{j=0}^{i-1} \boldsymbol{R}_{k+i}^{-\top} \boldsymbol{R}_{k+j}^+ \Delta t, i \geq 2, \#_{i,7} = \boldsymbol{R}_{c,i}^- \boldsymbol{\omega}_{c,i}^\times \boldsymbol{R}_{c,i}^{-\top}, i \geq 1$$

$$\#_{i,6} = \boldsymbol{R}_{k+i}^{-\top} \boldsymbol{v}_{k+i}^\times + \boldsymbol{\omega}_{c,k+i}^\times \boldsymbol{R}_{c,k+i}^{-\top} \boldsymbol{p}_{c,k+i}^{-\times}, i \geq 0$$

As the third column of $\boldsymbol{g}^\times$ is always 0, the first column of the $\hat{\boldsymbol{O}}_c$ suggests that ϕ_z is always non-observable despite the discrepancy between the operating point and real state. The position is also unobservable according to the third column.

When $\boldsymbol{\omega}_c$ is time-varying, the camera position is fully observable as multiple different measurements are considered. In this case, the last column of $\hat{\boldsymbol{O}}_c$ has full rank, but essentially not conflicting with $\boldsymbol{O}_c$. When $\boldsymbol{\omega}_c$ is strictly a non-zero constant for at least 7 consecutive time steps, the rank of the last column of $\hat{\boldsymbol{O}}_c$ is greater than $\boldsymbol{O}_c$. As the unobservable state depends on the real input that is inaccessible, it is impossible to design the operating points like [5] or [14]. However, this case is very rare as the robot motion is highly dynamical. Therefore, our filter will stay consistent with the underlying nonlinear system except for this rare case.

V. EXPERIMENTAL VALIDATION

We launched experiments on the Cassie bipedal robot to evaluate the proposed filter. The Cassie robot is equipped with an IMU and 14 encoders. The IMU can provide angular velocity and linear acceleration in robot-frame at 800 Hz. The encoders can provide joint angles and their derivatives are numerically computed at 2000Hz. Contact states are obtained via spring deflection measured by encoders and forward kinematics. A Intel Realsense T265 Tracking camera is mounted on the top of robot to provide velocity and angular velocity at 200Hz. Motion capture system is setup to obtain the ground truth.

The robot is walking over slippery terrain for about 4 meters within 35 seconds with the controller from [9] implemented. Snapshots of experiments are presented in Fig. 2.

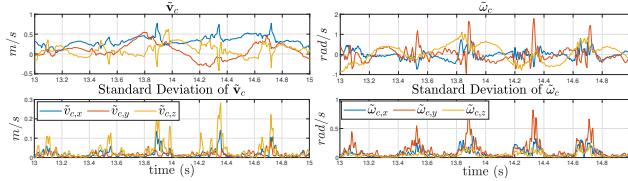Fig. 2: Cassie robot walking on slippery and unstable terrain.

Fig. 3: Tracking camera measurements & empirical std. dev. We use $n = 5$ in equation (62).

Tracking camera measurements and their empirical standard deviations are presented in Fig. 3. We also plotted the states estimated by fusion of only inertial and leg kinematics, denoted by legend "camera off" for comparison. We applied the outlier detection method [3] in both cases. The kinematics measurement with Mahalanobis distance greater than a threshold ρ, i.e $\chi^2 = \boldsymbol{\delta}^\top \boldsymbol{S}^{-1} \boldsymbol{\delta} > \rho$ will be discarded. As the slippage persisted for multiple seconds, the estimated value will diverge if we discard all the slippage when the camera is off. Thus, we carefully tune the threshold to maintain convergence while trying to discard as much outliers as possible. We use $\rho = 30.1$.

The estimated robot velocity and orientation in the robot-frame are presented in Fig. 4 and 5 respectively. The velocity in the robot-frame is observable thus converging fast. Rotation around the gravity axis, i.e, ϕ_z is not observable, thus the uncertainty slowly grows. ϕ_x and ϕ_y are observable and converges after a transient state. Due to outliers, the inertial-kinematics method inaccurately estimated some points but our method is always consistent with the ground truth.

Fig. 7 presents the estimated Cartesian position. As the position of robot is unobservable, a discrepancy is seen after an initial drift. However, the overall horizontal drift is smaller than 5% over the 3.5 m trajectory. The vertical drift is as small as 0.07 m, which is satisfactory compared to the InEKF in [10]. The orientation and position of camera in the robot-frame are presented in Fig. 6. All the 6 quantities converge and the 3σ covariance hull overlapped well with the ground truth. This result is quite satisfactory considering the highly time-varying camera measurement noise.

VI. CONCLUSIONS

This paper proposed a state estimation approach for legged robots by fusing inertial, velocity measurements from a tracking camera as well as from leg kinematics. The obtained
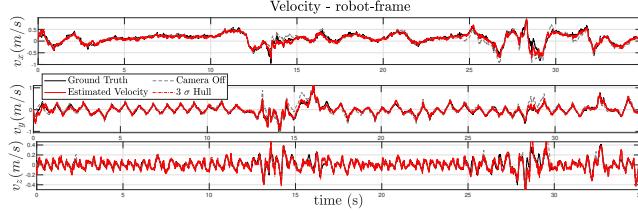

Fig. 4: Velocity estimation in robot-frame. The RMSE for v_x, v_y and v_z are 0.0703 m/s, 0.0660 m/s and 0.0513 m/s.
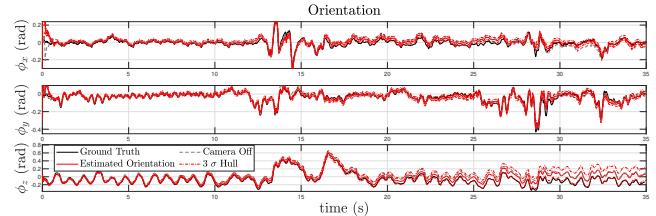

Fig. 5: Estimated robot orientation. The RMSE of ϕ_x, ϕ_y and ϕ_z are 0.0362 rad, 0.0213 rad and 0.185 rad respectively.
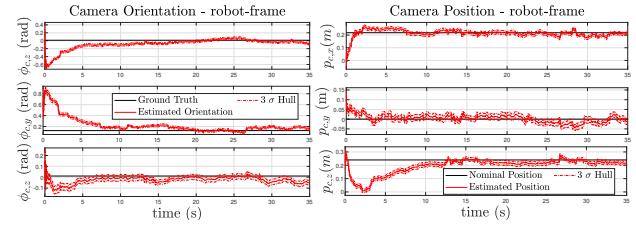

Fig. 6: Estimated camera pose. The values converge to the calibrated value after a transient state. The ground truth camera orientation is calibrated by aligning the robot and camera IMU. The nominal camera position is obtained by the CAD model of the platform.

information is fused by an invariant extended Kalman filter. The leg kinematics based velocity estimation is formulated as right-invariant observation. The misalignment between the camera and the robot-frame is also modeled thus enables auto-calibration of the camera pose. An online covariance tuning method is proposed to handle the highly time-varying measurement noise from the tracking camera. Nonlinear and discrete observability analysis suggest that only the rotation around the gravity vector and the absolute robot position is not observable, and our proposed filter remains consistent with the underlying nonlinear system.

Our method is evaluated in legged locomotion experiments with significant amount of slippage involved. The result suggests that our method could accurately estimate the robot inclination and the robot velocity with robustness. Although absolute velocity and rotation about gravity is unobservable, their drifts remains small. Camera pose is also successfully estimated by the filter.
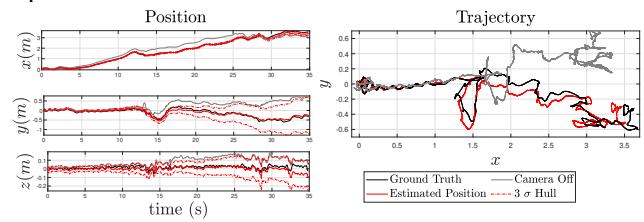
Fig. 7: Estimated robot position in 3D space. The estimated trajectory overlapped well with ground truth before the major slip at $x \approx 1.5m$. A shift is seen after the slip but the trajectory did not diverge. Note that the robot position is unobservable.

REFERENCES

[1] A. Barrau, "Non-linear state error based extended kalman filters with applications to navigation," Ph.D. dissertation, 2015.

[2] A. Barrau and S. Bonnabel, "The invariant extended kalman filter as a stable observer," *IEEE Transactions on Automatic Control*, vol. 62, no. 4, pp. 1797–1812, 2016.

[3] M. Bloesch, C. Gehring, P. Fankhauser, M. Hutter, M. A. Hoepflinger, and R. Siegwart, "State estimation for legged robots on unstable and slippery terrain," in *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*, Nov 2013, pp. 6058–6064.

[4] M. Bloesch, M. Burri, S. Omari, M. Hutter, and R. Siegwart, "Iterated extended kalman filter based visual-inertial odometry using direct photometric feedback," *The International Journal of Robotics Research*, vol. 36, no. 10, pp. 1053–1072, 2017.

[5] M. Bloesch, M. Hutter, M. A. Hoepflinger, S. Leutenegger, C. Gehring, C. D. Remy, and R. Siegwart, "State estimation for legged robots-consistent fusion of leg kinematics and imu," *Robotics*, vol. 17, pp. 17–24, 2013.

[6] M. Bloesch, S. Omari, P. Fankhauser, H. Sommer, C. Gehring, J. Hwangbo, M. A. Hoepflinger, M. Hutter, and R. Siegwart, "Fusion of optical flow and inertial measurements for robust egomotion estimation," in *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2014, pp. 3102–3107.

[7] M. Brossard, A. Barrau, and S. Bonnabel, "Ai-imu dead-reckoning," *IEEE Transactions on Intelligent Vehicles*, 2020.

[8] Z. Chen, K. Jiang, and J. C. Hung, "Local observability matrix and its application to observability analyses," in *[Proceedings] IECON'90: 16th Annual Conference of IEEE Industrial Electronics Society*. IEEE, 1990, pp. 100–103.

[9] Y. Gong, R. Hartley, X. Da, A. Hereid, O. Harib, J.-K. Huang, and J. Grizzle, "Feedback control of a cassie bipedal robot: Walking, standing, and riding a segway," in *2019 American Control Conference (ACC)*. IEEE, 2019, pp. 4559–4566.

[10] 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.

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

[12] R. Hartley, J. Mangelson, L. Gan, M. G. Jadidi, J. M. Walls, R. M. Eustice, and J. W. Grizzle, "Legged robot state-estimation through combined forward kinematic and preintegrated contact factors," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2018, pp. 1–8.

[13] R. Hermann and A. Krener, "Nonlinear controllability and observability," *IEEE Transactions on Automatic Control*, vol. 22, no. 5, pp. 728–740, 1977.

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

[15] F. Jenelten, J. Hwangbo, F. Tresoldi, C. D. Bellicoso, and M. Hutter, "Dynamic locomotion on slippery ground," *IEEE Robotics and Automation Letters*, vol. 4, no. 4, pp. 4170–4176, 2019.

[16] K. Kaneko, F. Kanehiro, S. Kajita, M. Morisawa, K. Fujiwara, K. Harada, and H. Hirukawa, "Slip observer for walking on a low friction floor," in *2005 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2005, pp. 634–640.

[17] E. Rosten, R. Porter, and T. Drummond, "Faster and better: A machine learning approach to corner detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 32, no. 1, pp. 105–119, Jan 2010.

[18] N. Rotella, M. Bloesch, L. Righetti, and S. Schaal, "State estimation for a humanoid robot," in *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2014, pp. 952–958.

[19] H. Takemura, M. Deguchi, J. Ueda, Y. Matsumoto, and T. Ogasawara, "Slip-adaptive walk of quadruped robot," *Robotics and Autonomous Systems*, vol. 53, no. 2, pp. 124–141, 2005.

[20] P. Varin and S. Kuindersma, "A constrained kalman filter for rigid body systems with frictional contact," in *International Workshop on the Algorithmic Foundations of Robotics (WAFR)*, 2018.

[21] D. Wisth, M. Camurri, and M. Fallon, "Robust legged robot state estimation using factor graph optimization," *IEEE Robotics and Automation Letters*, pp. 1–1, 2019.