

Multi-IMU Proprioceptive Odometry for Legged Robots

Shuo Yang, Zixin Zhang, Benjamin Bokser, Zachary Manchester

Abstract—This paper presents a novel, low-cost proprioceptive sensing solution for legged robots with point feet to achieve accurate low-drift long-term position and velocity estimation. In addition to conventional sensors, including one body Inertial Measurement Unit (IMU) and joint encoders, we attach an additional IMU to each calf link of the robot just above the foot. An extended Kalman filter is used to fuse data from all sensors to estimate the robot’s body and foot positions in the world frame. Using the additional IMUs, the filter is able to reliably determine foot contact modes and detect foot slips without tactile or pressure-based foot contact sensors. This sensing solution is validated in various hardware experiments, which confirm that it can reduce position drift by nearly an order of magnitude compared to conventional approaches with only a very modest increase in hardware and computational costs.

I. INTRODUCTION

Legged robots require precise estimation of physical states such as body position and orientation (pose), as well as body velocity, in order to balance, track velocity commands [1], and plan paths [2] on challenging terrains. In many scenarios, state estimation must be done solely with onboard sensors because external sensors such as GPS and motion-capture systems are not available.

A standard onboard sensing approach called Proprioceptive Odometry (PO) [3] uses an IMU in the robot’s body and several leg sensors (joint encoders and foot contact sensors) to estimate body pose (position and orientation) and velocity in a Kalman filter (KF) [1], [4], [5]. However, on resource-constrained legged robots equipped with cheap sensors, PO is known to have large position drift [4], usually over 10%, due to sensor noise and faulty foot-contact detection.

Recently published sensing solutions have added visual sensors such as cameras and lidar [6]–[8] to improve position estimation accuracy. With a camera or lidar onboard, position estimation drift can be reduced to as low as 0.5% [7]. However, this comes at a significant additional cost in terms of system complexity and computational power to process camera images or lidar point clouds. For low-cost lightweight legged robots with limited payload capacity, these visual sensing solutions may not be viable.

Our goal is to develop a PO solution that can achieve low position drift while requiring minimal additional hardware and computational resources. To achieve this, we propose Multi-IMU Proprioceptive Odometry (MIPO), a sensing solution that uses multiple IMUs and leg-joint encoders to significantly improve upon KF-based PO methods that only

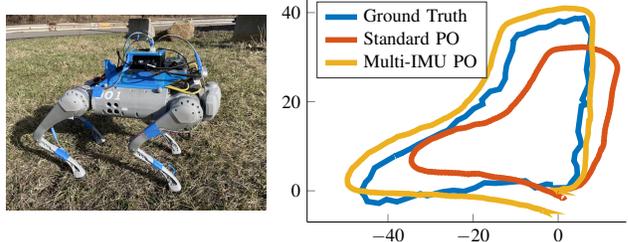


Fig. 1. **Left:** A Unitree Go1 robot equipped with foot IMUs. **Right:** Position estimates while walking over a 160m loop trajectory: Standard Proprioceptive Odometry (PO) has an average XY position drift of 18.4%, while our Multi-IMU PO solution achieves 2.59% average drift, a significant improvement achieved with minimal additional hardware and computational cost.

use a single IMU. Compared to visual sensors such as cameras and lidar, an IMU has much lower cost, energy consumption, and physical size. Therefore, we are able to add IMUs to the robot’s calf links near the feet with minimal impact on the overall design and cost. To fuse data from additional IMUs, we include world-frame foot positions and velocities in the state of an Extended Kalman Filter (EKF) and design a prediction model to update foot velocities using foot IMU data. More importantly, we also leverage the foot IMUs to detect contact and foot slip in a novel measurement model, overcoming a fundamental error source in conventional PO, where feet are assumed to have zero velocity relative to the ground while in contact [1], [4], [9], [10].

Our state-estimation approach is validated on a quadrupedal robot in a variety of conditions, and we make comparisons to several existing baseline methods. Our specific contributions include:

- A novel low-cost multi-IMU sensing solution for proprioceptive odometry on legged robots.
- An Extended Kalman Filter (EKF) with a prediction model that uses foot IMU data to update foot velocities and a measurement model that uses foot IMU data to determine foot contact modes and slip.
- Ablation studies and comparison experiments on hardware demonstrating significant reductions in position drift.

The remainder of this paper is organized as follows: Section II provides a review of related work. Section III presents the basics of legged robot state estimation. Section IV presents our main technical contributions. Section V compares our solution with baseline methods across a variety of experiments. An ablation study is also provided. Finally,

Authors are with the Robotics Institute and Department of Mechanical Engineering, Carnegie Mellon University, Pittsburgh, Pennsylvania, USA. Emails: {shuoyang, zixinz, bbokser, zmanches}@andrew.cmu.edu

Section VI concludes the paper.

II. RELATED WORK

Legged robot state estimation has received considerable attention in recent years [4], [8], [11]–[14]. With the increasing commercial availability of low-cost quadrupedal robots [15], [16], there is a strong need for cheap and reliable sensing solutions for such robots with limited computation resources.

One of the earliest legged robot state estimation methods using low-cost consumer-grade sensors was proposed in [4], where an Extended Kalman Filter (EKF) was used to combine measurements from a single IMU, joint encoders, and foot contact sensors. Since all sensors are contained within the body of the robot, this sensing solution is called Proprioceptive Odometry (PO). A simplified version of PO is used on the MIT Cheetah 3 robot [1], in which the state estimator only estimates body velocity and foot positions, so the filter has a linear form. To improve orientation estimation, the invariant EKF was developed in [17]. Although many variations on the EKF have been proposed, they all use the same basic types and number of sensors.

With proprioceptive sensors, velocity estimation is typically good enough for stable closed-loop control, but position drift is often as high as 10%-15% [4], [12], [17], [18]. A major source of error in PO comes from the assumption that a foot that is in contact with the ground has zero velocity [19]. However, in reality, the foot may be slipping [11], deforming [10], or rolling on the ground. Nevertheless, PO is widely used because not all legged robot control applications require high-accuracy position estimation.

In addition to proprioceptive sensors, cameras, and lidars are effective for state estimation in the Simultaneous Localization And Mapping (SLAM) research community [20], [21]. Vision-based sensors have also been successfully applied to legged robot state estimation: A loosely coupled visual-inertial leg odometry (VILO) solution using a tactical-grade IMU and high-quality cameras with an FPGA-based sensor synchronization mechanism to achieve less than 1% position estimation drift on the Boston Dynamics LS3 robot was built in [22]. A solution based on the factor graph [23] combines IMU, encoder, and camera measurements is presented in [13]. It is further improved by adding a lidar sensor to achieve 0.2%-0.4% drift performance [7]. The camera has also been shown to help eliminate some error sources in PO to improve performance [8], [10], [24]. However, because cameras and lidars generate high-bandwidth data, these methods are computationally intensive by nature.

Using multiple IMUs instead of one to improve estimation precision is common in many robotics research domains, such as pedestrian navigation systems [25], augmented and virtual-reality applications [26], and general visual-inertial odometry [27], where IMUs were installed on the same rigid body. Finally, an estimator is proposed in [28] using a network of IMUs installed on different parts of a humanoid robot for better joint-velocity sensing, but not for position estimation.

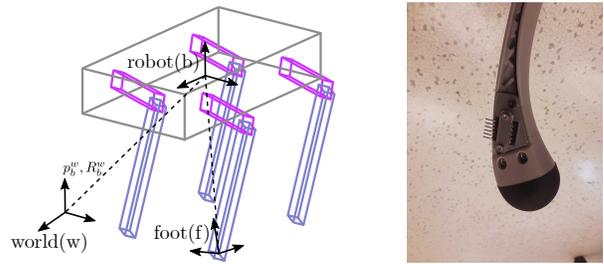


Fig. 2. **Left:** Important frame definitions. **Right:** The installation location of an IMU on one foot of a Unitree Go1 robot.

III. BACKGROUND

We now present the notation used throughout the paper and review conventional PO for legged robot state estimation. We use lowercase letters for scalars and frame abbreviations, boldface lowercase letters for vectors, and upper case letters for matrices and vector sets. The operation $[a; b; c]$ vertically concatenates elements a , b and c . The operator $[\mathbf{v}]^\times$ converts a vector $\mathbf{v} = [v_1; v_2; v_3] \in \mathbb{R}^3$ into the skew-symmetric “cross-product matrix,”

$$[\mathbf{v}]^\times = \begin{bmatrix} 0 & -v_3 & v_2 \\ v_3 & 0 & -v_1 \\ -v_2 & v_1 & 0 \end{bmatrix}, \quad (1)$$

such that $\mathbf{v} \times \mathbf{x} = [\mathbf{v}]^\times \mathbf{x}$.

A. Euler Angle Rotation

We parameterize the robot’s orientation using Tait–Bryan angles [1]. Specifically, $\boldsymbol{\theta} = [\theta_r; \theta_p; \theta_y]$ represents an orientation, where θ_y , θ_p , θ_r are commonly referred as yaw angle, pitch angle, and roll angle respectively. A formulation with quaternions [10] or other orientation parameters can also be easily derived. We denote $\cos(\theta_i) = c_i$ and $\sin(\theta_i) = s_i$ for $i \in \{y, p, r\}$. We can construct a rotation matrix from these angles:

$$R(\boldsymbol{\theta}) = \begin{bmatrix} c_p c_y & c_y c_p c_r - c_r c_y & c_r c_y + c_r c_y c_p \\ c_p c_y & c_r c_y + c_p c_r c_y & c_r c_p c_y - c_y c_r \\ -c_p & c_p c_r & c_p c_r \end{bmatrix}, \quad (2)$$

which transforms a vector represented in the robot body (b) frame to the world inertial (w) frame [29]. When necessary for clarity, we use superscripts and subscripts to explicitly show the frame transformation property of rotation matrices and the frame of a vector, so $R_b^w \cdot \mathbf{p}$ means the rotation matrix R transforms a vector \mathbf{p} represented in coordinate frame b into coordinate frame w . Important frames are shown in Fig. 2.

If the robot has an IMU installed at its center of mass, we denote its output angular velocity measurement as $\boldsymbol{\omega}_b$ and linear acceleration measurement as \mathbf{a}_b . Given the gravitational acceleration vector $\mathbf{g}_w = [0; 0; 9.8m/s^2]$, the robot acceleration in the world frame \mathbf{a}_w can be calculated as

$$\mathbf{a}_w = R(\boldsymbol{\theta})\mathbf{a}_b - \mathbf{g}_w. \quad (3)$$

The derivative of the yaw-pitch-roll angles $\dot{\boldsymbol{\theta}}$ is related to the robot body angular velocity $\boldsymbol{\omega}_b$ through the following

relationship:

$$\dot{\boldsymbol{\theta}} = \begin{bmatrix} \dot{\theta}_r \\ \dot{\theta}_p \\ \dot{\theta}_y \end{bmatrix} = \Omega(\boldsymbol{\theta}) \boldsymbol{\omega}_b = \begin{bmatrix} 1 & (s_p s_r)/c_p & (c_r s_p)/c_p \\ 0 & c_r & -s_r \\ 0 & s_r/c_p & c_r/c_p \end{bmatrix} \boldsymbol{\omega}_b. \quad (4)$$

B. Forward Kinematics & Leg Odometry Velocity

In this section, we review forward kinematics and describe how to infer the robot's body velocity from joint-angle information. For the j th leg of a legged robot, we define $\boldsymbol{\phi}$ as a vector containing all joint angles. The forward kinematics function is denoted $\mathbf{p}_f = g(\boldsymbol{\phi}) \in \mathbb{R}^3$, whose output is the foot position in the robot body frame. The derivative of this equation leads to the Jacobian matrix $J(\boldsymbol{\phi})$ that maps $\dot{\boldsymbol{\phi}}$ into the foot's linear velocity in the body frame:

$$\mathbf{v}_f = \dot{\mathbf{p}}_f = J(\boldsymbol{\phi}) \dot{\boldsymbol{\phi}}. \quad (5)$$

Let \mathbf{p}_f^w denote the foot position in the world frame. It is a function of the robot's body position \mathbf{p} and joint angles $\boldsymbol{\phi}$:

$$\mathbf{p}_f^w = \mathbf{p} + R_b^w \mathbf{p}_f = \mathbf{p} + R_b^w g(\boldsymbol{\phi}). \quad (6)$$

A fundamental assumption used in PO is the zero-velocity assumption for a non-slipping foot in contact, which means $\dot{\mathbf{p}}_f^w = \mathbf{0}$. As shown in [10], with this assumption we can derive an expression for the body velocity in the world frame:

$$\mathbf{v} = -R_b^w [J(\boldsymbol{\phi}) \dot{\boldsymbol{\phi}} + [\boldsymbol{\omega}]^\times g(\boldsymbol{\phi})]. \quad (7)$$

This velocity is called the Leg Odometry (LO) velocity, because its integration is the body displacement [30].

C. Extend Kalman Filtering

The Extended Kalman Filter (EKF) is widely used in robot state estimation. The basic building blocks are a dynamics or "process" model

$$\mathbf{x}_{k+1} = f(\mathbf{x}_k) + \mathbf{n}_k \quad (8)$$

and a measurement model

$$\mathbf{y}_k = h(\mathbf{x}_k) + \mathbf{w}_k, \quad (9)$$

where \mathbf{n}_k and \mathbf{w}_k are zero-mean additive noise terms drawn from Gaussian distributions with covariances Σ_n and Σ_w , respectively. We also make use of the model Jacobians $F_k = \partial f / \partial \mathbf{x}_k$ and $H_k = \partial h / \partial \mathbf{x}_k$ [31].

Given an estimated state $\hat{\mathbf{x}}_k$ at time k and a sensor measurement vector $\bar{\mathbf{y}}_k$, the EKF first "predicts" the mean of the state distribution at time $k+1$ using the process model $\hat{\mathbf{x}}_{k+1|k} = f(\hat{\mathbf{x}}_k)$. The filter then performs a "measurement update", where a measurement residual $\mathbf{z}_{k+1} = \bar{\mathbf{y}}_k - h(\hat{\mathbf{x}}_{k+1|k})$ and the Kalman gain [5] K_{k+1} are calculated, so that

$$\hat{\mathbf{x}}_{k+1} = \hat{\mathbf{x}}_{k+1|k} + K_{k+1} \mathbf{z}_{k+1}, \quad (10)$$

is the state at time $k+1$. The covariance of the state is also updated accordingly. We refer the reader to standard references on state estimation for more details [31].

D. Standard Single-IMU Proprioceptive Odometry

Let the robot's state be $\mathbf{x} = [\mathbf{p}; \mathbf{v}; \boldsymbol{\theta}; \mathbf{s}_1, \dots, \mathbf{s}_j, \dots, \mathbf{s}_L]$, where $\mathbf{p} \in \mathbb{R}^3$ is the robot position in the world frame, $\boldsymbol{\theta}$ is

the robot's orientation Tait-Bryan angles, and $\mathbf{v} \in \mathbb{R}^3$ is the linear velocity of the robot's body represented in the world frame. For $j \in \{1, \dots, L\}$ where L is the total number of legs of the robot, \mathbf{s}_j is the foot position of leg j represented in the world frame. For clarity, we will only discuss the case when $L=1$ in this section and drop the symbol j from subsequent equations. The robot's sensors generate a number of measurements including IMU linear acceleration \mathbf{a}_b , IMU angular velocity $\boldsymbol{\omega}_b$, joint angle $\boldsymbol{\phi}$, joint-angle velocity $\dot{\boldsymbol{\phi}}$, and c which is a binary contact flag with $c=1$ indicating foot contact.

Standard PO uses the EKF to estimate the state from a single IMU and the LO velocity [1], [4]. The IMU is biased [32] and the LO velocity may also have a bias due to leg kinematic parameter changes [10]. We do not address these biases in this work, but they can be easily added to the EKF using well-known techniques to improve the overall estimation accuracy [31].

The process model of standard PO is based on IMU kinematics. A discrete-time dynamics update using Euler integration is presented in [1],

$$\hat{\mathbf{x}}_{k+1} = \begin{bmatrix} \hat{\mathbf{p}}_{k+1} \\ \hat{\mathbf{v}}_{k+1} \\ \hat{\boldsymbol{\theta}}_{k+1} \\ \hat{\mathbf{s}}_{k+1} \end{bmatrix} = \begin{bmatrix} \hat{\mathbf{p}}_k + \Delta t \hat{\mathbf{v}}_k \\ \hat{\mathbf{v}}_k + \Delta t (R(\hat{\boldsymbol{\theta}}_k) \mathbf{a}_b - \mathbf{g}_w) \\ \hat{\boldsymbol{\theta}}_k + \Delta t (\Omega(\hat{\boldsymbol{\theta}}_k) \boldsymbol{\omega}_b) \\ \hat{\mathbf{s}}_k \end{bmatrix}, \quad (11)$$

where Δt is the time interval between k and $k+1$.

A common heuristic is used to reflect the fact that, during foot swing, we cannot update the foot position in the process model [4]: For the term corresponding to $\hat{\mathbf{s}}$ in \mathbf{n}_k , we set its covariance σ_s to a large value if $c=0$, and a small value otherwise:

$$\sigma_s = c \sigma_c + (1-c) \sigma_n. \quad (12)$$

σ_c and $\sigma_n \gg \sigma_c$ are all tunable hyper-parameters.

We formulate the EKF measurement model following [1]. From sensor measurements, a vector $\bar{\mathbf{y}}_k$ is obtained as

$$\bar{\mathbf{y}}_k = \begin{bmatrix} g(\boldsymbol{\phi}) \\ -J(\boldsymbol{\phi}) \dot{\boldsymbol{\phi}} + [\boldsymbol{\omega}_b]^\times g(\boldsymbol{\phi}) \end{bmatrix} \quad (13)$$

The measurement function $h(\hat{\mathbf{x}}_k)$ is defined as

$$h(\hat{\mathbf{x}}_k) = \begin{bmatrix} R(\hat{\boldsymbol{\theta}}_k)^T (\hat{\mathbf{s}}_k - \hat{\mathbf{p}}_k) \\ R(\hat{\boldsymbol{\theta}}_k)^T \hat{\mathbf{v}}_k \end{bmatrix} \quad (14)$$

The first term of the residual $\bar{\mathbf{y}}_k - h(\hat{\mathbf{x}}_k)$ indicates that the estimated body position and foot position must differ by a distance equal to the leg forward kinematics position. The second term ensures that the estimated robot body velocity matches the LO velocity (7), which we refer to as a "zero-velocity" observation model. Subsequently, we can utilize residuals from all legs in the EKF as shown in equation (10). However, measurement residuals are only applicable for non-slipping standing feet. Therefore, the noise covariance Σ_w is adjusted based on the contact flag c as in equation (12) [4].

IV. MULTI-IMU PROPRIOCEPTIVE ODOMETRY

The standard single-IMU PO method presented in Section III-D has two fundamental limitations: First, foot positions cannot be updated during the foot’s swing phase due to the noise adjustment mechanism (12). If the robot has a fully airborne phase, then leg sensors will have no contribution to the state estimate. Second, the zero-velocity assumption used when deriving (7) is seldom true on hardware. The LO velocity always underestimates the true velocity if the foot is rolling during contact, as shown in Fig. 3. Both of these limitations can be addressed by adding additional IMUs to the robot’s feet. We refer to this sensor-and-algorithm combination as Multi-IMU Proprioceptive Odometry (MIPO).

We revise the estimator state to $\mathbf{x} = [\mathbf{p}; \mathbf{v}; \boldsymbol{\theta}; \mathbf{s}_j; \dot{\mathbf{s}}_j]$ where we explicitly include foot velocity $\dot{\mathbf{s}}_j$ in the state. Again, we consider $L = 1$ and drop $j \in \{1 \dots, L\}$ for brevity. We assume all sensors are synchronized and produce data at the same frequency. In addition to sensor measurements introduced in Section III-D, we also get \mathbf{a}_f and $\boldsymbol{\omega}_f$, the foot acceleration reading and the foot angular velocity reading from an IMU installed on the foot in the foot frame.

A. EKF Process Model

We now use foot acceleration to predict foot velocity. The process model is changed from (11) to

$$\hat{\mathbf{x}}_{k+1} = \begin{bmatrix} \hat{\mathbf{p}}_{k+1} \\ \hat{\mathbf{v}}_{k+1} \\ \hat{\boldsymbol{\theta}}_{k+1} \\ \hat{\mathbf{s}}_{k+1} \\ \hat{\dot{\mathbf{s}}}_{k+1} \end{bmatrix} = \begin{bmatrix} \hat{\mathbf{p}}_k + \Delta t \hat{\mathbf{v}}_k \\ \hat{\mathbf{v}}_k + \Delta t (R(\hat{\boldsymbol{\theta}}_k) \mathbf{a}_b - \mathbf{g}_w) \\ \hat{\boldsymbol{\theta}}_k + \Delta t (\Omega(\hat{\boldsymbol{\theta}}_k) \boldsymbol{\omega}_b) \\ \hat{\mathbf{s}}_k + \Delta t \dot{\hat{\mathbf{s}}} \\ \dot{\hat{\mathbf{s}}}_k + \Delta t (R(\hat{\boldsymbol{\theta}}_k) R_f^b(\boldsymbol{\phi}) \mathbf{a}_f - \mathbf{g}_w) \end{bmatrix}. \quad (15)$$

The rotation matrix $R_f^b(\boldsymbol{\phi})$ is a function of joint angles $\boldsymbol{\phi}$, and can be obtained from forward kinematics [29]. It maps the acceleration in the foot frame to the body frame. Compared to (11), the noise covariance of this process model does not depend on the contact flag because the dynamics are continuous. The foot velocity noise covariance is constant as long as IMUs do not saturate their readings, which can be guaranteed by proper controller design and IMU hardware selection.

B. The Pivoting Contact Model

For legged robots with spherical “point” feet, at any instant in time, the robot body is pivoting around the contact point, regardless of whether the contact foot is stationary or rolling. Thus, the contact foot’s linear velocity should equal the cross product of the foot angular velocity vector $\boldsymbol{\omega}$ and a pivoting vector \mathbf{d} pointing from the contact point to the body center, as illustrated in Fig. 3. $\boldsymbol{\omega}$ and \mathbf{d} are calculated as

$$\boldsymbol{\omega}(\hat{\mathbf{x}}_k, \boldsymbol{\phi}, \boldsymbol{\omega}_f) = R(\hat{\boldsymbol{\theta}}_k) R_f^b(\boldsymbol{\phi}) \boldsymbol{\omega}_f, \quad (16)$$

$$\mathbf{d}(\hat{\mathbf{x}}_k, \boldsymbol{\phi}) = -d_0 \cdot \mathbf{n} / \|\mathbf{n}\|, \quad (17)$$

where d_0 is the distance between the foot center and the foot surface, and $\mathbf{n} = R(\hat{\boldsymbol{\theta}}_k) \mathbf{g}(\boldsymbol{\phi})$ is the contact normal vector expressed in the world frame. This “pivoting” model captures

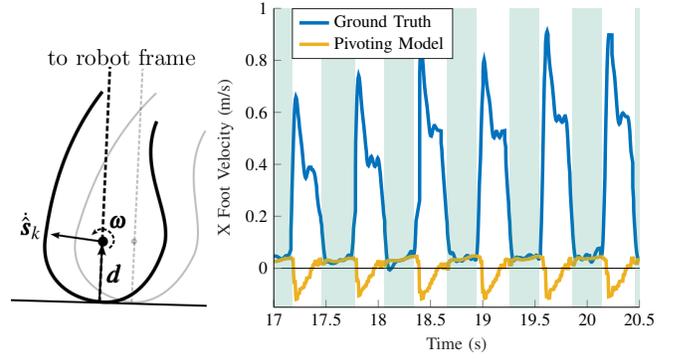


Fig. 3. **Left:** Illustration of the pivoting model for a foot that has rolling contact with the ground. The estimated foot velocity $\dot{\mathbf{s}}_k$ depends on $\boldsymbol{\omega}$ and \mathbf{d} , as defined in (16) and (17). **Right:** Comparison of ground-truth foot velocity captured with a motion capture system to estimation using the pivoting model (18). When the foot has non-zero rolling velocity during contacts (shaded regions), the pivoting model agrees with the ground truth velocity very well while the zero-velocity model treats foot velocity as zero.

rolling contact better than [10] and only relies on the foot IMU.

C. EKF Measurement Model

Using the pivoting model, we revise the measurement model as follows:

$$h(\hat{\mathbf{x}}_k, \boldsymbol{\phi}, \boldsymbol{\omega}_f) = \begin{bmatrix} R(\hat{\boldsymbol{\theta}}_k)^T (\hat{\mathbf{s}}_k - \hat{\mathbf{p}}_k) \\ R(\hat{\boldsymbol{\theta}}_k)^T (\hat{\mathbf{v}}_k - \dot{\hat{\mathbf{s}}}_k) \\ \hat{\mathbf{s}}_k - \boldsymbol{\omega} \times \mathbf{d} \end{bmatrix}. \quad (18)$$

The first term is the same as in (14). The second term comes from (6) without assuming $\mathbf{v}_f^w = 0$. In contrast to (14), these two terms do not have varying measurement noise since they stay the same across foot contact switches.

We refer to the last term, which is based on the pivoting model, as a pivoting measurement. Because this measurement term is an implicit function of both states and sensor measurements $h(\mathbf{x}, \mathbf{y}) = 0$ rather than an explicit function $\mathbf{y} = h(\mathbf{x})$, its corresponding value in the measurement vector $\bar{\mathbf{y}}_k$ is always zero [33]. Note that the pivoting constraint is only valid when a foot is in contact with the ground.

D. Foot Contact and Slip Detection

We replace the covariance-scaling heuristic (12) with a statistical test based on Mahalanobis distance [11]:

$$\sqrt{\mathbf{z}^T \mathbf{S}^{-1} \mathbf{z}} < \sigma, \quad (19)$$

where σ is a hyperparameter, $\mathbf{z} = \dot{\hat{\mathbf{s}}} - \boldsymbol{\omega} \times \mathbf{d}$, and \mathbf{S} is its corresponding covariance matrix calculated in the Kalman filter. If (19) is satisfied, we treat the foot as being in non-slipping contact and include the corresponding pivoting measurement in the update (18).

Since the foot velocity is drastically different in the swing and stance phases (see Fig. 3 unshaded regions), (19) can distinguish foot phases without relying on a contact sensor. Similar mechanisms for the zero-velocity model have been used in [4] and [12], and we show, through an ablation study

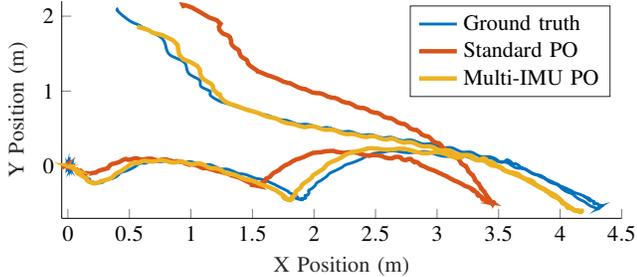


Fig. 4. The estimated XY position trajectory comparison of the standard PO and MIPO. The total trajectory length is 10.5m. The standard PO estimation drifts 11.39% on average, and its maximum RSE is 1.04m. While the result of MIPO has 2.31% average drift and 0.25m maximum RSE.

| | Frequency | Solve Time | Median Drift |
|-------------|-----------|------------|--------------|
| Standard PO | 200Hz | 1.81ms | 11.05% |
| MIPO | 200Hz | 2.50ms | 2.61% |

TABLE I
PERFORMANCE SUMMARY

in Section V, that the statistical test is also important for the pivoting model to achieve accurate position estimation.

V. EXPERIMENTS

We conducted a series of indoor and outdoor experiments on hardware to compare our MIPO with the baseline method, standard single-IMU PO [1], and Cerberus Visual-Inertial-Leg Odometry (VILO) [8]. We focused on evaluating their position estimation performance, especially XY position. Common evaluation metrics used in the state-estimation literature include Root-Square-Error (RSE) and drift percentage. RSE is defined as the Euclidean distance between the ground truth and the estimated position at a given time instance. Drift percentage is defined as the ratio of the RSE to the total distance traveled. The data and code used to generate results are available on Github¹.

A. Sensor Hardware Design

The MIPO hardware does not significantly alter the form factor of the Go1 robot. The Go1 robot’s built-in proprietary MEMS IMU sensor and joint motor encoder data can be obtained at 200Hz via Ethernet. The robot also has four pressure contact sensors on the feet that can be thresholded to obtain binary contact flags for the baseline method. Our MIPO, however, does not use contact sensor data. Instead, four MPU9250 IMUs are installed on each foot, right above the foot, as shown in Fig. 2. Since it is difficult to directly install an IMU at the exact foot center, we transform foot IMU outputs to the foot center frame using the transformation measured in the CAD model. An Arduino Teensy board communicates with the foot IMUs, acquiring their outputs at 200Hz. An Intel NUC mini computer finally collects sensor data from the Go1 robot and the Teensy board to run

¹<https://github.com/ShuoYangRobotics/Multi-IMU-Proprioceptive-Odometry>

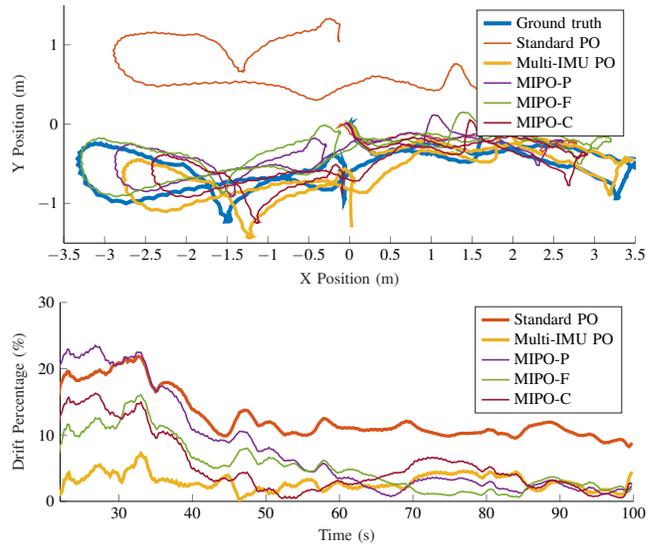


Fig. 5. **Top:** XY position trajectory estimation results of the standard PO, MIPO, and three MIPO variants used in the ablation study. The total trajectory length is 21.5m. **Bottom:** The drift percentages over time of all methods. MIPO has the smallest (4.21%), followed by MIPO-F (7.75%), MIPO-C (11.3%), MIPO-P (12.04%), and the standard PO (16.87%).

the MIPO algorithm. The computer also runs a nonlinear predictive control locomotion controller [34].

B. Position Estimation Evaluation

We first compare the results of MIPO against standard PO in an indoor environment. The robot operates in a lab space equipped with a motion capture (MoCap) system which provides the ground truth pose. The robot uses the trotting gait or flying trotting gait (with a full airborne phase between leg switching) to locomote in arbitrary directions with a speed of 0.4-1.0 m/s on flat ground. Fig. 4 compares the estimated position trajectories. Table I compares their per-loop solve times and average drifts across five different datasets. MIPO has larger state dimensions so the computation time is slightly longer, but the time is still within the budget (5 ms). Moreover, the drift percentage is significantly lower.

In outdoor environments where neither MoCap or reliable GPS signals are available, we use the Cerberus [8] VILO algorithm as ground truth to compare MIPO and standard PO. As can be seen in Fig. 1, MIPO achieves a much smaller position drift than standard PO after traveling 160m over varying terrains.

C. Ablation Study

In this section, we study the individual contribution of the pivoting model and the statistical test introduced in Section IV-C. We create three algorithm variants: 1) MIPO-P, where the last term in (18) is \hat{s}_k instead of $\hat{s}_k - \omega \times d$. We vary its measurement noise according to the contact flag but do not perform the statistical test (19). In this way, MIPO-P is essentially a standard PO that uses MIPO’s process model. 2) MIPO-F, which adopts the pivoting model from MIPO, but differs by replacing the statistical test with the measurement noise adjustment mechanism based on the contact flag. And

3) MIPO-C, which uses MIPO's process model, MIPO-P's measurement model, and the statistical test on the last measurement term instead of varying measurement noise. The results of standard PO, MIPO, and all three variants are shown in Fig. 5.

MIPO-F has performance closer to MIPO than standard PO, MIPO-P, and MIPO-C, showing that the largest performance contributor is the pivoting constraint. This result suggests that the zero-velocity model fundamentally limits the capability of standard PO to achieve low-drift position estimation, even if more accurate contact-flag generation methods can be used to avoid contact-detection errors in standard PO.

VI. CONCLUSIONS & FUTURE WORK

We have presented the Multi-IMU Proprioceptive Odometry (MIPO), a legged robot state estimation solution with IMUs in both the body and feet. Experiments have shown that additional IMUs in feet can significantly reduce position drift and improve overall estimation accuracy while keeping computation and hardware costs low. Moreover, MIPO provides an alternative method for detecting foot contact modes and foot slip without using contact sensors. It is a compelling replacement for conventional single-IMU PO. The process and measurement models in MIPO can be easily added to other EKF-based PO methods [1], [35]. MIPO's accuracy should further improve if IMU biases [4], kinematic parameters [10], and other calibration errors are addressed. Moreover, the contact flag generated by MIPO may also benefit downstream control algorithms.

REFERENCES

- [1] G. Bledt, M. J. Powell, B. Katz, J. Di Carlo, P. M. Wensing, and S. Kim, "Mit cheetah 3: Design and control of a robust, dynamic quadruped robot," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2018, pp. 2245–2252.
- [2] J. Norby and A. M. Johnson, "Fast global motion planning for dynamic legged robots," in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, pp. 3829–3836.
- [3] V. Vasilopoulos, O. Arslan, A. De, and D. E. Koditschek, "Sensor-based legged robot homing using range-only target localization," in *2017 IEEE International Conference on Robotics and Biomimetics (ROBIO)*. IEEE, 2017, pp. 2630–2637.
- [4] M. Bloesch, M. Hutter, M. A. Hoepflinger, S. Leutenegger, C. Gehring, C. D. Remy, and R. Siegwart, "State estimation for legged robots - consistent fusion of leg kinematics and imu," *Robotics*, vol. 17, pp. 17–24, 2013.
- [5] R. E. Kalman, "A new approach to linear filtering and prediction problems," *Journal of Basic Engineering*, vol. 82, no. 1, 1960.
- [6] 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.
- [7] D. Wisth, M. Camurri, and M. Fallon, "Vilens: Visual, inertial, lidar, and leg odometry for all-terrain legged robots," *IEEE Transactions on Robotics*, 2022.
- [8] S. Yang, Z. Zhang, Z. Fu, and Z. Manchester, "Cerberus: Low-drift visual-inertial-leg odometry for agile locomotion," in *2023 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2023.
- [9] R. Hartley, M. G. Jadidi, J. W. Grizzle, and R. M. Eustice, "Contact-aided invariant extended kalman filtering for legged robot state estimation," *arXiv preprint arXiv:1805.10410*, 2018.
- [10] 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.
- [11] M. Bloesch, C. Gehring, P. Fankhauser, M. Hutter, M. A. Hoepflinger, and R. Siegwart, "State estimation for legged robots on unstable and slippery terrain," in *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2013, pp. 6058–6064.
- [12] J.-H. Kim, S. Hong, G. Ji, S. Jeon, J. Hwangbo, J.-H. Oh, and H.-W. Park, "Legged robot state estimation with dynamic contact event information," *IEEE Robotics and Automation Letters*, vol. 6, no. 4, pp. 6733–6740, 2021.
- [13] D. Wisth, M. Camurri, and M. Fallon, "Robust legged robot state estimation using factor graph optimization," *IEEE Robotics and Automation Letters*, vol. 4, no. 4, pp. 4507–4514, 2019.
- [14] 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 *International Conference on Robotics and Automation*. IEEE, 2018.
- [15] Unitree, "A1," <https://www.unitree.com/products/a1/>, 2023.
- [16] B. Dynamics, "Spot," <https://www.bostondynamics.com/spot>, 2023.
- [17] 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.
- [18] D. Wisth, M. Camurri, and M. Fallon, "Preintegrated velocity bias estimation to overcome contact nonlinearities in legged robot odometry," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2020, pp. 392–398.
- [19] 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*, vol. 2, no. 2, pp. 1023–1030, 2017.
- [20] J. Zhang and S. Singh, "Loam: Lidar odometry and mapping in real-time," in *Robotics: Science and Systems*, vol. 2, no. 9, 2014.
- [21] R. Mur-Artal, J. M. M. Montiel, and J. D. Tardos, "Orb-slam: a versatile and accurate monocular slam system," *IEEE Transactions on Robotics*, vol. 31, no. 5, pp. 1147–1163, 2015.
- [22] J. Ma, M. Bajracharya, S. Susca, L. Matthies, and M. Malchano, "Real-time pose estimation of a dynamic quadruped in gps-denied environments for 24-hour operation," *The International Journal of Robotics Research*, vol. 35, no. 6, pp. 631–653, 2016.
- [23] F. Dellaert, M. Kaess, et al., "Factor graphs for robot perception," *Foundations and Trends® in Robotics*, vol. 6, no. 1-2, 2017.
- [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*, vol. 7, no. 2, pp. 4456–4463, 2022.
- [25] J. B. Bancroft and G. Lachapelle, "Data fusion algorithms for multiple inertial measurement units," *Sensors*, vol. 11, no. 7, 2011.
- [26] A. Jadid, L. Rudolph, F. Pankratz, and G. Klinker, "Utilizing multiple calibrated imus for enhanced mixed reality tracking," in *2019 IEEE International Symposium on Mixed and Augmented Reality Adjunct (ISMAR-Adjunct)*. IEEE, 2019, pp. 384–386.
- [27] K. Eckenhoff, P. Geneva, and G. Huang, "Mimic-vins: A versatile and resilient multi-imu multi-camera visual-inertial navigation system," *IEEE Transactions on Robotics*, vol. 37, no. 5, pp. 1360–1380, 2021.
- [28] X. Xinjilefu, "State estimation for humanoid robots," Ph.D. dissertation, 2015.
- [29] R. M. Murray, Z. Li, and S. S. Sastry, *A mathematical introduction to robotic manipulation*. CRC press, 2017.
- [30] P.-C. Lin, H. Komsuoglu, and D. E. Koditschek, "A leg configuration measurement system for full-body pose estimates in a hexapod robot," *IEEE Transactions on Robotics*, vol. 21, no. 3, pp. 411–422, 2005.
- [31] P. S. Maybeck, *Stochastic Models, Estimation, and Control*. Academic Press, 1982, vol. 1.
- [32] R. T. Savely, "Apollo experience report: onboard navigational and alignment software," 1972.
- [33] D. Simon, "Kalman filtering with state constraints: a survey of linear and nonlinear algorithms," *IET Control Theory & Applications*, vol. 4, no. 8, pp. 1303–1318, 2010.
- [34] F. Farshidian et al., "OCS2: An open source library for optimal control of switched systems," [Online]. Available: <https://github.com/leggedrobotics/ocs2>.
- [35] 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 *International Conference on Intelligent Robots and Systems*, 2018, pp. 3783–3790.