2019 International Conference on Robotics and Automation (ICRA)
Palais des congres de Montreal, Montreal, Canada, May 20-24, 2019

Improving the Robustness of Visual-Inertial Extended Kalman Filtering

James Jackson[1], Jerel Nielsen[1], Tim McLain[1], Randal Beard[1]

***Abstract*— Visual-inertial navigation methods have been shown to be an effective, low-cost way to operate autonomously without GPS or other global measurements, however most filtering approaches to VI suffer from observability and consistency problems. To increase robustness of the state-of-the-art methods, we propose a three-fold improvement. First, we propose the addition of a linear drag term in the velocity dynamics which improves estimation accuracy. Second, we propose the use of a partial-update formulation which limits the effect of linearization errors in partially-observable states, such as sensor biases. Finally, we propose the use of a keyframe reset step to enforce observability and consistency of the normally unobservable position and heading states. While all of these concepts have been used independently in the past, our experiments demonstrate additional strength when they are used simultaneously in a visual-inertial state estimation problem.**

In this paper, we derive the proposed filter and use a Monte Carlo simulation experiment to analyze the response of visual-inertial Kalman filters with the above described additions. The results of this study show that the combination of all of these features significantly improves estimation accuracy and consistency.

I. Introduction

Visual-inertial (VI) navigation is becoming an increasingly important tool for autonomous operation of miniature aerial vehicles (MAVs) and other robotic agents. While many missions can be performed using GPS or other global measurements to constrain drift, there are numerous scenarios that do not have reliable access to these global measurements. For example, a camera and MEMS IMU can provide a low-cost way to autonomously navigate, and visual camera features provide a method to constrain IMU drift, while also making sensor biases observable for accurate integration.

Recent results in this area have demonstrated remarkable performance and capability [1], [2], [3], [4], [5], [6]. While smoothing methods and nonlinear batch optimization-based methods [7], [8], [9] have demonstrated significant advantages in terms of accuracy and consistency, they can be too computationally intense for many low-cost platforms. Filtering approaches have the advantage of being computationally efficient but can struggle in certain situations, due to significant nonlinearities and unobservability [10], [11], [12]. This paper discusses filtering techniques for VI estimation that significantly increase robustness to these issues.

One major source of unobservability in VI filtering is the parameterization of feature locations. Feature locations parameterized in an inertial coordinate frame typically assume observability of the transform to that frame. In many situations, however, this transform is unobservable, and estimation becomes inconsistent [10], [11], [12]. Recent methods have shown how to estimate features in the camera frame, rather than an inertial frame [2]. These parameterizations partition the states cleanly into observable and non-observable states, with global position and heading being completely unobservable. The unobservability of position and heading can be handled using the method proposed by [10], where the position and heading states are periodically reset, so that they remain observable and consistent. The global state and uncertainty are then calculated using other methods such as batch optimization, which are external to the Kalman filter.

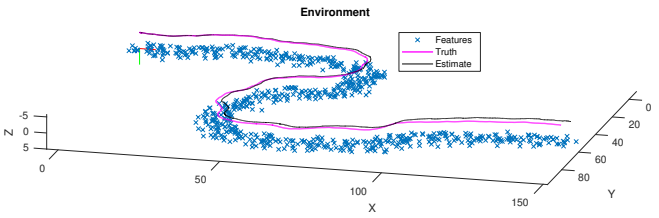

Fig. 1. A sample trajectory from the Monte Carlo simulation experiment.

Finally, many visual-inertial estimation approaches assume no knowledge about certain aspects of the system dynamics. In some applications, knowledge of specific parts of the system dynamics can help improve estimation accuracy and prevent divergence in certain modes at the expense of becoming less portable to other systems. [13], [14] For example, information regarding the speed capabilities of a multirotor aircraft can bound changes in estimates of depth to visual features. Leishman et al. [14] showed that including a linear model of drag on a multirotor significantly improves estimation accuracy. We will use this model to improve estimator robustness in this work.

Another source of nonlinearity and unobservability is the presence of filter states that are only partially observable or unobservable given specific vehicle motion. Examples of these states include IMU biases, depth to features and the above mentioned linear drag term. Brink [15] has shown that using a partial update can improve filter robustness to these so-called nuisance states, while maintaining consistency.

In this paper, we extend the robocentric visual-inertial Kalman filtering approach described in [2] with the principles of relative navigation described in [16]. We also show that improving the dynamic model can significantly improve estimation accuracy of VI estimation applied to a multirotor and use the partial update formulation to deal with the additional nuisance state used in modeling drag. The paper is organized as follows. In section II, we describe several mathematical concepts and notation used throughout the paper. In section III, we briefly discuss the derivation of our baseline filter [2] with the improved dynamic model.

Sections III-B and III-C discuss the measurement models used and sections III-D and III-E detail the keyframe reset and partial update steps, respectively. Finally, section IV details a Monte Carlo simulation experiment and compares the performance of the proposed improvements in terms of accuracy and consistency.

II. Notation

The following definitions are used throughout the paper.

$\mathbf{e}_i$	Unit vector with a one in the i^{th} element
$\mathbf{p}_{b/I}^I$	Position of the body, with respect to the world frame, expressed in the world frame
$\mathbf{v}_{b/I}^b$	Velocity of the body frame, with respect to the world frame, expressed in the body frame
$\mathbf{q}_I^b$	Quaternion describing rotation from the world frame to the body frame
$\boldsymbol{\beta}_a$	Accelerometer bias
$\boldsymbol{\beta}_\omega$	Rate gyro bias
b	Linear drag coefficient
$\boldsymbol{\zeta}_{i/c}^c$	Unit vector directed at the i^{th} feature from the camera origin, expressed in the camera frame
$\mathbf{q}_c^{\zeta_i}$	Quaternion which describes the rotation from the camera $\mathbf{e}_3$ axis to the unit vector $\boldsymbol{\zeta}_{i/c}^c$
ρ_i	Inverse distance to the i^{th} feature

We will also make extensive use of the skew-symmetric matrix operator defined by

$$\mathbf{v}^\wedge \triangleq \begin{bmatrix} 0 & -v_3 & v_2 \\ v_3 & 0 & -v_1 \\ -v_2 & v_1 & 0 \end{bmatrix},$$

that is related to the cross-product between two vectors with

$$\mathbf{v} \times \mathbf{w} = \mathbf{v}^\wedge \mathbf{w}.$$

To convert back to a vector from a skew-symmetric matrix, we use the $\cdot^\vee$ operator, so that

$$\left(\mathbf{v}^\wedge\right)^\vee = \mathbf{v}.$$

A. Quaternions

We will use Hamiltonian notation for unit quaternions $\in \mathcal{S}^3$

$$\mathbf{q} = q_0 + q_x\mathbf{e}_1 + q_y\mathbf{e}_2 + q_z\mathbf{e}_3 = \begin{bmatrix} q_0 \\ \bar{\mathbf{q}} \end{bmatrix}, \tag{1}$$

which defines the passive rotation matrix based on a unit quaternion as

$$R\left(\mathbf{q}\right) = \left(2q_0^2 - 1\right) I - 2q_0\bar{\mathbf{q}}^\wedge + 2\bar{\mathbf{q}}\bar{\mathbf{q}}^\top \in SO(3). \tag{2}$$

This definition results in $R_a^b\mathbf{r}^a$ being interpreted as the original vector $\mathbf{r}^a$ expressed in the new coordinate frame b.

The exponential mapping for a unit quaternion is defined as

$$\begin{aligned} \exp &: \mathfrak{so}(3)^\vee \sim \mathbb{R}^3 \to \mathcal{S}^3 \\ \exp\left(\boldsymbol{\delta}\right) &\triangleq \begin{bmatrix} \cos\left(\frac{\|\boldsymbol{\delta}\|}{2}\right) \\ \sin\left(\frac{\|\boldsymbol{\delta}\|}{2}\right)\frac{\boldsymbol{\delta}}{\|\boldsymbol{\delta}\|} \end{bmatrix}, \end{aligned} \tag{3}$$

with the corresponding logarithmic map defined as

$$\begin{aligned} \log &: \mathcal{S}^3 \to \mathfrak{so}(3)^\vee \cong \mathbb{R}^3 \\ \log\left(\mathbf{q}\right) &\triangleq 2\,\mathrm{atan2}\left(\|\bar{\mathbf{q}}\|, q_0\right)\frac{\bar{\mathbf{q}}}{\|\bar{\mathbf{q}}\|}. \end{aligned} \tag{4}$$

The notion of computing the difference between two group elements leads to defining uncertainty over a member of the Lie manifold. For example, the attitude quaternion $\mathbf{q}_I^b$ has four elements but only three degrees of freedom, so its covariance should be a 3×3 matrix. Using the logarithmic map, we can define the attitude covariance as

$$E\left[\log\left(\left(\hat{\mathbf{q}}_I^b\right)^{-1} \otimes \mathbf{q}_I^b\right)\log\left(\left(\hat{\mathbf{q}}_I^b\right)^{-1} \otimes \mathbf{q}_I^b\right)^\top\right] \in \mathbb{R}^{3\times 3}. \tag{5}$$

Eq. (5) is significant because the covariance is parameterized in the Lie algebra $\mathfrak{so}\left(3\right)$ (which is a vector space) of $SO\left(3\right)$ and therefore, can be used in a Kalman filtering framework.

B. $\boxplus$ and $\boxminus$ operators

Hertzberg et al. [17] describe a new syntax that simplifies working with Lie groups in a filtering and optimization framework by introducing the $\boxplus$ and $\boxminus$ operators. This syntax allows us to work with elements of Lie groups in a notation similar to that of vectors and will be used to describe our filter derivation. The $\boxplus$ and $\boxminus$ operators are defined differently for different groups. For $\mathbb{R}^n$, they are simply defined as the typical addition and subtraction operations. For attitude quaternions $\in \mathcal{S}^3$, these operators are defined by

$$\begin{aligned} \boxplus &: \mathcal{S}^3 \times \mathbb{R}^3 \to \mathcal{S}^3 \\ &\quad \mathbf{q} \boxplus \boldsymbol{\theta} \triangleq \mathbf{q} \otimes \exp\left(\boldsymbol{\theta}\right) \\ \boxminus &: \mathcal{S}^3 \times \mathcal{S}^3 \to \mathbb{R}^3 \\ &\quad \mathbf{q} \boxminus \mathbf{p} \triangleq \log\left(\mathbf{p}^{-1} \otimes \mathbf{q}\right). \end{aligned}$$

One common application of this syntax can be seen below in the discretized quaternion dynamics. With $\boldsymbol{\theta} = \boldsymbol{\omega}_{b/I}^b\,dt$, we have

$$\mathbf{q}_I^b\left(t + dt\right) = \mathbf{q}_I^b\left(t\right) \boxplus \boldsymbol{\theta} \tag{6}$$

$$\boldsymbol{\theta} = \mathbf{q}_I^b\left(t + dt\right) \boxminus \mathbf{q}_I^b\left(t\right). \tag{7}$$

While this syntax is convenient, it is important to note that the dimensionality of $\boldsymbol{\theta}$ and $\mathbf{q}_I^b$ are different in this case. The quaternion is not a vector and has four parameters, while $\boldsymbol{\theta}$ has only three parameters but exists in a vector space.

C. Feature Bearing Parameterization

As in [2], we parameterize the feature bearing states in the camera frame as rotations $\mathbf{q}_c^{\zeta_i} \in \mathcal{S}^3 \sim R_c^{\zeta_i} \in SO\left(3\right)$, which describe the rotation from the camera $\mathbf{e}_3$ axis to the unit vector directed at the feature. The unit vector directed at feature i with respect to the camera frame c is then defined by

$$\boldsymbol{\zeta}_{i/c}^c = \left(R_c^{\zeta_i}\right)^\top \mathbf{e}_3 \in \mathcal{S}^2 \subset \mathbb{R}^3, \tag{8}$$

where we can see that this simply expresses the direction of the feature in the camera frame.

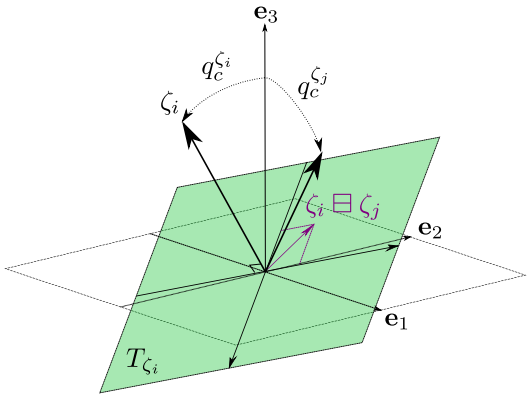

Fig. 2. Illustration of feature bearing vector geometry.

The difference between two unit vectors $\boldsymbol{\zeta}_i \boxminus \boldsymbol{\zeta}_j$ can be described using axis-angle representation, where the direction of the axis of rotation is orthogonal to both of the unit vectors, and its length is scaled by the magnitude of rotation, as shown in Figure 2. There are actually only two degrees of freedom in this parameterization because rotation about either feature vector does not change unit vector direction. To remove the redundant degree of freedom, we note that the axis of shortest rotation is always in the plane normal to $\boldsymbol{\zeta}_{i/c}^c$ and define a projection matrix

$$T_{\zeta_i} = \left(R_c^{\zeta_i}\right)^\top \begin{bmatrix}\mathbf{e}_1 & \mathbf{e}_2\end{bmatrix} \in \mathbb{R}^{3\times 2}, \tag{9}$$

which reduces the dimensionality of the axis-angle representation to this plane. It can be seen that this projection matrix is just the two basis vectors orthogonal to feature direction, defined in the camera reference frame.

We must then define the $\boxplus$ and $\boxminus$ operators associated with feature bearing vectors as

$$\begin{aligned}\boxplus &: SO(3) \times \mathbb{R}^2 \to SO(3)\\ &\quad \mathbf{q}_c^\zeta \boxplus \boldsymbol{\delta} \triangleq \exp(T_\zeta \boldsymbol{\delta}) \otimes \mathbf{q}_c^\zeta\\ \boxminus &: SO(3) \times SO(3) \to \mathbb{R}^2\\ &\quad \mathbf{q}_c^{\zeta_j} \boxminus \mathbf{q}_c^{\zeta_i} \triangleq \theta T_{\zeta_i}^\top \mathbf{s},\end{aligned}$$

where the axis $\mathbf{s}$ and angle θ between the two feature direction vectors are given by

$$\begin{aligned}\theta &= \cos^{-1}\left(\boldsymbol{\zeta}_i^\top \boldsymbol{\zeta}_j\right)\\ \mathbf{s} &= \frac{\boldsymbol{\zeta}_i \times \boldsymbol{\zeta}_j}{\|\boldsymbol{\zeta}_i \times \boldsymbol{\zeta}_j\|}.\end{aligned}$$

With only two degrees of freedom, and with all feature vectors referencing the camera $\mathbf{e}_3$ axis, there are an infinite number of unit quaternions which can be used to represent the same unit vector. The difference between these rotations is some angle of rotation about the bearing vector itself. This is removed by the projection operation and can therefore be neglected. Reference [18] explores more deeply the validity of the $\boxplus$ and $\boxminus$ operators under this assumption.

III. Derivation

In this section, we derive the relevant geometry and dynamics to fully describe and implement the filter proposed in this paper.

A. State Definition and Kinematics

Let the state $\mathbf{x} \in \mathbb{R}^6 \times \mathcal{S}^3 \times \mathbb{R}^7 \times \mathcal{S}^3 \times \mathbb{R} \times \cdots \times \mathcal{S}^3 \times \mathbb{R}$ be defined by

$$\begin{aligned}\mathbf{x} = \Big[\ \mathbf{p}_{b/I}^I \quad \mathbf{v}_{b/I}^b \quad \mathbf{q}_I^b \quad \boldsymbol{\beta}_a \quad \boldsymbol{\beta}_\omega \quad b \quad \mathbf{q}_c^{\zeta_1} \quad \rho_1\\ \cdots \quad \mathbf{q}_c^{\zeta_n} \quad \rho_n \ \Big],\end{aligned} \tag{10}$$

with n tracked features. The corresponding covariance matrix P is then defined as

$$P = E\left[(\mathbf{x} \boxminus \hat{\mathbf{x}})(\mathbf{x} \boxminus \hat{\mathbf{x}})^\top\right] \in \mathbb{R}^{(16+3n)\times(16+3n)},$$

where $\boxminus$ for objects composed of multiple group elements implies the use of the appropriate $\boxminus$ operator for each element.

Given measured acceleration $\bar{\mathbf{a}}_{b/I}^b$ and measured angular velocity $\bar{\boldsymbol{\omega}}_{b/I}^b$, the state has kinematics $\dot{\mathbf{x}} = f(\mathbf{x}, \mathbf{u} + \boldsymbol{\eta})$ with the elements of f given by [14] and defined as

$$\begin{aligned}\dot{\mathbf{p}}_{b/I}^I &= \left(R_I^b\right)^\top \mathbf{v}_{b/I}^b\\ \dot{\mathbf{v}}_{b/I}^b &= \mathbf{e}_3\mathbf{e}_3^\top \mathbf{a}_{b/I}^b + R_I^b \mathbf{g}^I - b\left(I - \mathbf{e}_3\mathbf{e}_3^\top\right)\mathbf{v}_{b/I}^b - \left(\boldsymbol{\omega}_{b/I}^b\right)^\wedge \mathbf{v}_{b/I}^b\\ \dot{\mathbf{q}}_I^b &= \boldsymbol{\omega}_{b/I}^b\\ \dot{\boldsymbol{\beta}}_a &= \mathbf{0}\\ \dot{\boldsymbol{\beta}}_\omega &= \mathbf{0}\\ \dot{b} &= \mathbf{0}\\ \dot{\mathbf{q}}_c^{\zeta_i} &= -T_{\zeta_i}^\top\left(\boldsymbol{\omega}_{c/I}^c + \rho_i\left(\boldsymbol{\zeta}_{i/c}^c\right)^\wedge \mathbf{v}_{c/I}^c\right)\\ \dot{\rho}_i &= \rho_i^2\left(\boldsymbol{\zeta}_{i/c}^c\right)^\top \mathbf{v}_{c/I}^c,\end{aligned} \tag{11}$$

where b is a linear drag term [14], $\mathbf{u} = \begin{bmatrix}\mathbf{a}_{b/I}^b & \boldsymbol{\omega}_{b/I}^b\end{bmatrix}$ is the input, $\boldsymbol{\eta} = [\boldsymbol{\eta}_a \quad \boldsymbol{\eta}_\omega]$ is input noise, and

$$\begin{aligned}\mathbf{a}_{b/I}^b &= \bar{\mathbf{a}}_{b/I}^b - \boldsymbol{\beta}_a - \boldsymbol{\eta}_a\\ \boldsymbol{\omega}_{b/I}^b &= \bar{\boldsymbol{\omega}}_{b/I}^b - \boldsymbol{\beta}_\omega - \boldsymbol{\eta}_\omega.\end{aligned}$$

Camera linear and angular velocities are also given by

$$\begin{aligned}\mathbf{v}_{c/I}^c &= R_b^c\left(\mathbf{v}_{b/I}^b + \left(\boldsymbol{\omega}_{b/I}^b\right)^\wedge \mathbf{p}_{c/b}^b\right)\\ \boldsymbol{\omega}_{c/I}^c &= R_b^c \boldsymbol{\omega}_{b/I}^b,\end{aligned}$$

where R_b^c is the fixed rotation from body to camera frame and $\mathbf{p}_{c/b}^b$ is the fixed translation from body to camera in the body frame.

In the proposed filter, we employ the typical continuous-discrete Extended Kalman Filter (EKF) equations. However, the use of $\boxplus$ and $\boxminus$ operators requires a slightly different treatment of the propagation and update equations. We propagate the filter forward in time and apply discrete updates

according to

$$\hat{\mathbf{x}}(t+dt) = \hat{\mathbf{x}}(t) \boxplus f(\hat{\mathbf{x}}(t), \mathbf{u}(t))\, dt$$
$$\hat{\mathbf{x}}^{+} = \hat{\mathbf{x}} \boxplus K(\mathbf{z} \boxminus h(\hat{\mathbf{x}})),$$

where K is the Kalman gain, $\mathbf{z}$ is a measurement, and $h(\hat{\mathbf{x}})$ is a measurement model.

B. Camera Measurement Model

Given a pixel measurement (u, v), pixel location of the camera's optical axis (u_0, v_0), camera focal lengths (f_x, f_y), and relative landmark location in the camera frame, the pinhole camera model may be written in terms of $\mathbf{x}$ as

$$h_{cam}(\mathbf{x}) = \frac{1}{\mathbf{e}_3^\top \boldsymbol{\zeta}^c} \begin{bmatrix} f_x & 0 & 0 \\ 0 & f_y & 0 \end{bmatrix} \boldsymbol{\zeta}^c + \begin{bmatrix} u_0 \\ v_0 \end{bmatrix}. \tag{12}$$

The Jacobian $\partial h_{cam}/\partial \mathbf{x}$ of the camera measurement model is given by

$$H_{cam} = \begin{bmatrix} \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & 0 & H_1 & 0 & \cdots & H_n & 0 \end{bmatrix},$$

where using the chain rule, we have

$$H_i = \frac{1}{\mathbf{e}_3^\top \boldsymbol{\zeta}_{i/c}^c} \begin{bmatrix} f_x & 0 & 0 \\ 0 & f_y & 0 \end{bmatrix} \left(\frac{\boldsymbol{\zeta}_{i/c}^c \mathbf{e}_3^\top}{\mathbf{e}_3^\top \boldsymbol{\zeta}_{i/c}^c} - I_{3\times 3} \right) \left(\boldsymbol{\zeta}_{i/c}^c \right)^{\wedge} T_{\zeta_i}.$$

C. Accelerometer Measurement Model

Using the multirotor drag model from [14] in (11) provides the benefit that velocity becomes directly observed by the accelerometer (assuming a linear drag constant). It is assumed that the accelerometer measures total acceleration of the body, neglecting gravity, in addition to a constant bias $\boldsymbol{\beta}_a$ and zero-mean white noise $\boldsymbol{\eta}_a$. If we also assume that thrust T acts only along the body $\mathbf{e}_3$ axis, we can consider just the body $\mathbf{e}_1$ and $\mathbf{e}_2$ axes, removing any dependence of the measurement on T. The measurement model is then given by

$$h_{acc}(\mathbf{x}) = I_{2\times 3} \left(-b \mathbf{v}_{b/I}^b + \boldsymbol{\beta}_a + \boldsymbol{\eta}_a \right). \tag{13}$$

The Jacobian $\partial h_{acc}/\partial \mathbf{x}$ is given by

$$H_{acc} = \begin{bmatrix} \mathbf{0} & -bI_{2\times 3} & \mathbf{0} & I_{2\times 3} & \mathbf{0} & -I_{2\times 3}\mathbf{v}_{b/I}^b & \mathbf{0} & \cdots \end{bmatrix}.$$

D. Keyframe Reset

As shown in [10] and [16], performing a keyframe reset when global states are unobservable can dramatically improve filter consistency and accuracy. A keyframe reset is performed by resetting the global position and heading states to zero and updating the covariance matrix appropriately. Each reset step results in a new *node* being declared in a pose graph structure, which can then incorporate loop closures and other measurements as part of a global optimization routine. Figure 3 shows an illustration of the coordinate frames involved in the keyframe reset. Here, we note that our setup slightly differs from [10] and [16] in that there is no altimeter measurement available, so altitude is also unobservable, and we must also reset that state. Therefore, we can see in Figure 3 that node frames are co-located with keyframes, instead of on a ground plane.

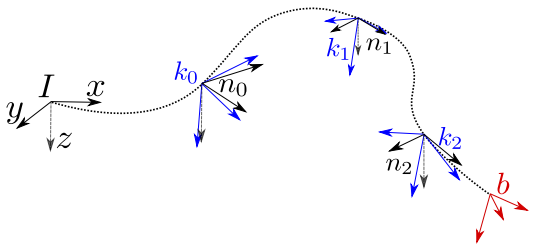

Fig. 3. Keyframes k_i are declared at periodic intervals along the trajectory flown by the MAV, while node frames n_i are associated with each keyframe and are gravity-aligned but co-located with each keyframe. The current body frame b is estimated with respect to the most recent keyframe. New keyframes are declared when less than 25 percent of the features present in the previous keyframe are still present. This promotes observability of the transform between b and the most recent keyframe.

E. Partial Update

A common difficulty faced in visual-inertial navigation is the estimation of nuisance states which may only be partially observable during many maneuvers. In the filter derived in this paper, these states include the inverse depth to each feature ρ_i, accelerometer and gyro biases $\boldsymbol{\beta}_a$ and $\boldsymbol{\beta}_\omega$, and the linear drag term b. As noted in [15], estimating these terms in the traditional manner can cause filter divergence but ignoring them or considering them as known constants may produce an overconfident estimate. Because of the abundance of these states in our system, we employ a version of the partial-update Schmidt-Kalman filter proposed by [15]. This method allows the designer to tune the effect of a measurement update on the i^{th} state with a scalar gain γ_i, while correctly estimating uncertainty in these partially-updated states. While this method loses optimality guarantees in estimating these states in a linear Kalman-filtering framework, it has been shown to speed up convergence of these nuisance states by limiting the effect of linearization errors when applied to the non-linear IMU-camera extrinsics estimation problem [15].

A drawback of the formulation given in [15] is the intermediate calculation of $\hat{\mathbf{x}}^{+}$and P^{+}. We can manipulate these equations to remove this intermediate calculation and maintain algebraic equivalence. Let us first define $\lambda_i = 1 - \gamma_i$, and for N states, also define

$$\boldsymbol{\lambda} = \begin{bmatrix} \lambda_1 & \lambda_2 & \cdots & \lambda_N \end{bmatrix},$$

which contains our tuning parameters. The values in this vector range from zero to one with ones indicating a full update to those particular states. The state and covariance updates may now be given by

$$\hat{\mathbf{x}}^{++} = \hat{\mathbf{x}}^{-} \boxplus \left(\boldsymbol{\lambda} \odot K \left(\mathbf{z} \boxminus \mathbf{h}\left(\hat{\mathbf{x}}^{-} \right) \right) \right)$$
$$P^{++} = P^{-} + \Lambda \odot \Big((I - KH) P^{-} (I - KH)^\top + KRK^\top - P^{-} \Big),$$

where we've employed the numerically stable Joseph form

of the covariance update, $\odot$ is the Hadamard product, and

$$\mathbf{1} = [1 \quad 1 \quad \cdots \quad 1]^\top$$
$$\Lambda = \mathbf{1}\boldsymbol{\lambda}^\top + \boldsymbol{\lambda}\mathbf{1}^\top - \boldsymbol{\lambda}\boldsymbol{\lambda}^\top.$$

IV. Results

To identify improvements to consistency and accuracy, we employed a Monte Carlo (MC) simulation of a MAV with a nonlinear aerodynamic model. The multirotor was commanded to fly approximately five meters above a simulated ground plane at a constant forward velocity of one meter per second. The commanded heading for each iteration evolved according to a random walk. A fourth-order Runge Kutta integration scheme was used for the truth comparison. A sample trajectory is shown in Figure 1.

Camera measurements consisted of static landmarks projected onto a simulated image plane via the pin-hole camera model and were corrupted by a small amount of white noise. Landmarks were chosen by randomly selecting enough features in the camera's field of view to fill the state vector. These same features were then selected in subsequent time steps until they left the camera's field of view, at which point another landmark was randomly generated in the field of view. This removes any dependence on a feature tracker in the MC simulation and results in ideal performance because there are no data association errors. However, this approach is appropriate for filter comparisons in an MC simulation because we wish to identify differences in filter performance under ideal conditions. Accelerometer and gyro measurements were corrupted with Gauassian noise and slowly varying biases similar to the observed noise in hardware experiments.

We implemented four different filters for comparison. The *baseline* (BL) filter is the same filter derived in [2] except with the measurement model for features given as (12) rather than the patch-based model in the original work. This was primarily done to simplify modeling in the simulation environment and to guarantee that all filters received the same measurements. The second filter modifies the baseline with a linear drag term (DT) as shown in (13), while the third filter modifies the baseline with keyframe resets (KF) given in Section III-D. The fourth filter augments the baseline with a drag term, keyframe reset, and a partial update (KF+DT+PU). Each of these filters were given identical inputs and measurements for each MC iteration, and the relevant process and sensor noise covariance matrices used in each filter were derived from the corresponding simulation parameters.

Inverse depth to each feature was initialized using the recommended values in [19] of $\rho_0 = {}^1\!/_{2d_{min}}$ and $R_0 = {}^1\!/_{16d_{min}}$ with a minimum distance to each feature assumed to be $d_{min} = 2$ meters. To deal with negative depth estimates, we used the method in [20], where any negative depth estimates were immediately re-initialized to d_{min} and the covariance appropriately expanded to account for the additional uncertainty. Because keyframes are not tied to a specific image in this estimator (as opposed to the implementation in [16]) new keyframes were declared when more than one half of the features present at the declaration of the previous keyframe were lost.

Absolute accuracy of each filter was compared using the root mean squared error (RMSE) of the position and attitude states. Because the filters with a keyframe reset step estimate this transform with respect to a local keyframe, each time a new keyframe was declared, (or each time a new node was created) both the true state $\mathbf{x}^n$ and the estimated state $\hat{\mathbf{x}}^n$ of each filter were saved, even in the filters with no keyframe reset step. We then calculated the RMSE of the estimated relative transform (position and attitude) between the previously declared node frame and the current body frame T_n^b for each filter

$$\begin{aligned} J_{RMS} &= \left\| \hat{T}_n^b \boxminus T_n^b \right\| \\ &= \left\| \begin{bmatrix} \hat{\mathbf{p}}_{b/n}^n - \mathbf{p}_{b/n}^n \\ \hat{\mathbf{q}}_n^b \boxminus \mathbf{q}_n^b \end{bmatrix} \right\|. \end{aligned}$$

This method not only ensures that we perform a fair comparison between filters, but it also ensures that the sometimes large heading errors accumulated before accelerometer and gyroscope bias measurements converge do not confound RMSE calculations later on in the trajectory.

Filter consistency was analyzed using normalized estimator error squared (NEES) or the Mahalanobis distance of the position and attitude states. Because NEES is weighted by the current covariance matrix of each estimator, the NEES of a filter with a keyframe reset is calculated with respect to relative pose, while the NEES of a filter without a keyframe reset is calculated with respect to global pose. Therefore, NEES is calculated according to

$$\epsilon = \left\{ \begin{array}{ll} \left(\hat{T}_n^b \boxminus T_n^b\right)^\top P_{T_n^b} \left(\hat{T}_n^b \boxminus T_n^b\right) & \text{if KF} \\ \left(\hat{T}_I^b \boxminus T_I^b\right)^\top P_{T_I^b} \left(\hat{T}_I^b \boxminus T_I^b\right) & \text{otherwise} \end{array} \right\}.$$

Because NEES is calculated over the transform states with 6-DOF (position and attitude), a histogram of the NEES of an ideal filter should fit a χ^2 distribution with six degrees of freedom and remain constant over time.

We performed 2016 MC iterations of a five-minute simulation study and calculated the RMSE and NEES at each time step (250 Hz). The average RMSE and NEES over time for each filter in the MC simulation study are shown in Figure 6. In this plot, we see that the RMSE of each filter decreases as each filter evolves in time and converges on the unknown biases. A histogram of the RMSE and NEES for each estimator at the final time is given in Figure 7.

It is clear from the results of this study that using keyframe resets dramatically affects RMSE and NEES, resulting more accurate and consistent pose estimates. In filters without a keyframe reset step, the unobservable position and heading states cause the filter to become increasingly inconsistent over time, resulting in large linearization errors and suboptimal sensor fusion [10].

It appears that while the drag term improves pose accuracy, it degrades consistency. This is not altogether unexepected as

the drag term is only partially observable and the resulting linearization error on the drag term measurement update (13) causes the filter to become overconfident. The improved accuracy, however comes from better state integration which arises from the improved dynamic model.

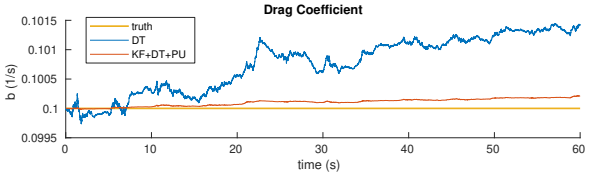

Fig. 4. Drag term estimates of a single MC iteration with and without the partial update.

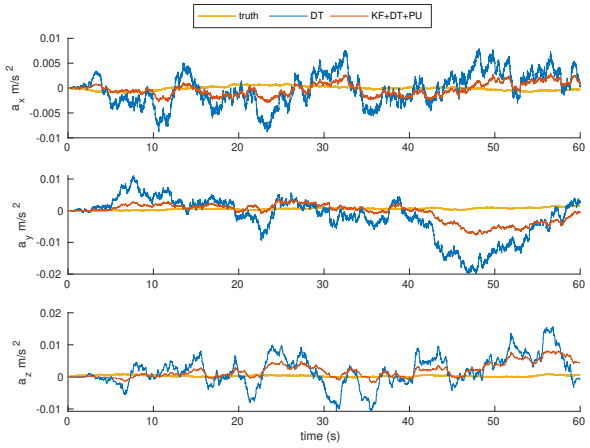

Fig. 5. Accelerometer biases of a single MC iteration with and without the partial update.

The overconfidence caused by the drag term can be mitigated by using a partial update. In the (KF+DT+PU) filter, γ_b was set to 0.02, which reduced the effect of linearization error on the state and covariance. Figure 4, shows a single run of the drag term with and without the partial update. In this plot, the drag term without the partial update produces oscillations corresponding to changes in attitude. This is most certainly incorrect as we have no reason to believe that the constant drag term should be correlated with attitude. The partial update attenuates these oscillations and allows us to benefit from the improved dynamic model. A similar effect is observed in accelerometer and gyroscope bias estimates. We see in Figure 5, that without the drag term, acclerometer bias estimates become strongly correlated with attitude. Again, the partial update damps this oscillatory response and keeps the estimate more aligned with truth.

V. Conclusions

We have shown that augmenting visual-inertial extended Kalman filtering with keyframe resets, an improved dynamic model, and partial updates greatly improves accuracy and consistency in VI filtering. This is clearly demonstrated in Figures 6 and 7. The use of keyframe resets improves filter consistency and accuracy without any observed negative

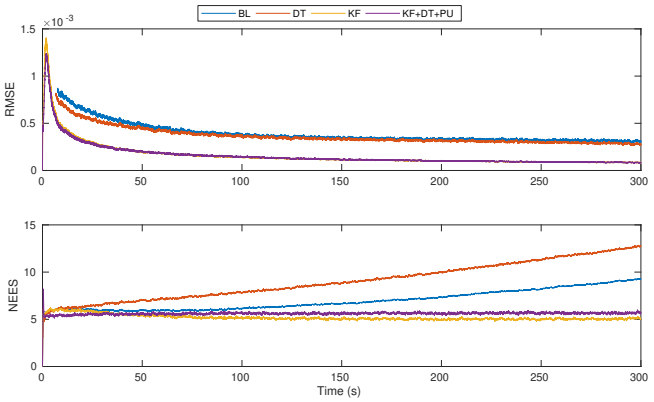

Fig. 6. Average RMSE of the transform from the most recent keyframe (top) and average NEES (bottom) for each filter over the entire simulation time over 2016 runs.

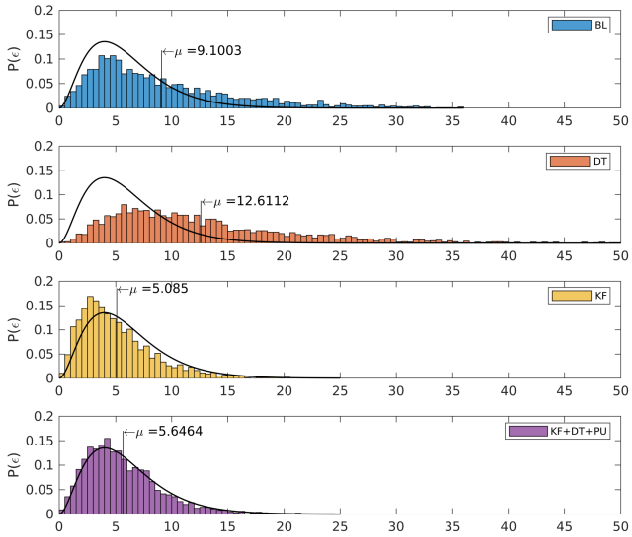

Fig. 7. The χ^2 distribution with six degrees of freedom compared against each filter at the final simulation time of 5 minutes using 2016 samples.

consequences. Augmenting the dynamic model with a linear drag term also improves accuracy but at the expense of degraded consistency. This inconsistency can be directly mitigated through the use of a partial update, thus, providing better accuracy from the improved dynamic model, while maintaining filter consistency. Finally, the combination of all three proposed improvements was shown to improve filter accuracy and consistency over the baseline filter.

References

[1] Michael Bloesch, Michael Burri, Sammy Omari, Marco Hutter, and Roland Siegwart. Iterated extended Kalman filter based visual-inertial odometry using direct photometric feedback. *International Journal of Robotics Research*, 36(10):1053–1072, 2017.

[2] Michael Bloesch, Sammy Omari, Marco Hutter, and Roland Siegwart. Robust Visual Inertial Odometry Using a Direct EKF-Based Approach. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Hamburg, Germany, 2015. IEEE.

[3] Anastasios I. Mourikis and Stergios I. Roumeliotis. A multi-state constraint Kalman filter for vision-aided inertial navigation. *Proceedings - IEEE International Conference on Robotics and Automation*, pages 3565–3572, 2007.

[4] Christian Forster, Zichao Zhang, Michael Gassner, Manuel Werlberger, and Davide Scaramuzza. Semi-Direct Visual Odometry for Monocular , Wide-angle, and Muti-Camera Systems. *IEEE Transactions on Robotics*, 33:249–265, 2017.

[5] Christian Forster, Matia Pizzoli, Davide Scaramuzza, and A Motivation. Fast Semi-Direct Monocular Visual Odometry. *Proceedings - IEEE International Conference on Robotics and Automation*, pages 15–22, 2014.

[6] Shaojie Shen, Nathan Michael, and Vijay Kumar. Tightly-coupled monocular visual-inertial fusion for autonomous flight of rotorcraft MAVs. *Proceedings - IEEE International Conference on Robotics and Automation*, 2015-June(June):5303–5310, 2015.

[7] Frank Dellaert. Factor Graphs and GTSAM : A Hands-on Introduction. (September):1–27, 2012.

[8] Tong Qin, Peiliang Li, and Shaojie Shen. VINS-Mono: A Robust and Versatile Monocular Visual-Inertial State Estimator. *arXiv preprint arXiv:1708.03852*, 2017.

[9] Zhenfei Yang and Shaojie Shen. Monocular visual-inertial state estimation with online initialization and camera-IMU extrinsic calibration. *IEEE Transactions on Automation Science and Engineering*, 14(1):39–51, 2017.

[10] David O Wheeler, Daniel P Koch, James S Jackson, Tim W McLain, and Randal W Beard. Relative navigation: A keyframe-based approach for observable gps-degraded navigation. *IEEE Control Systems Magazine*, 38(4):30–48, Aug 2018.

[11] Joel A. Hesch, Dimitrios G. Kottas, Sean L. Bowman, and Stergios I. Roumeliotis. Consistency analysis and improvement of vision-aided inertial navigation. *IEEE Transactions on Robotics*, 30(1):158–176, 2014.

[12] Joan Solà. Consistency of the monocular EKF-SLAM algorithm for three different landmark parametrizations. *Proceedings - IEEE International Conference on Robotics and Automation*, (1):3513–3518, 2010.

[13] Michael Burri, M Dätwiler, Markus Achtelik, and Roland Siegwart. Robust state estimation for micro aerial vehicles based on system dynamics. volume 2015, pages 5278–5283, 06 2015.

[14] Robert C. Leishman, John C. MacDonald, Randal W. Beard, and Timothy W. McLain. Quadrotors and accelerometers: State estimation with an improved dynamic model. *IEEE Control Systems*, 34(1):28–41, 2014.

[15] Kevin M Brink. Partial-Update Schmidt–Kalman Filter. *Journal of Guidance, Control, and Dynamics*, 40(9):2214–2228, 2017.

[16] Daniel P Koch, David O Wheeler, Randal W Beard, Tim W McLain, and Kevin M Brink. Relative Multiplicative Extended Kalman Filter for Observable GPS-Denied Navigation. *All Faculty Publications*, (1963), 2017.

[17] Christoph Hertzberg, René Wagner, Udo Frese, and Lutz Schröder. Integrating generic sensor fusion algorithms with sound state representations through encapsulation of manifolds. *Information Fusion*, 14(1):57–77, 2013.

[18] Michael Bloesch, Michael Burri, Sammy Omari, Marco Hutter, and Roland Siegwart. Iterated extended Kalman filter based visual-inertial odometry using direct photometric feedback. *International Journal of Robotics Research*, 36(10):1053–1072, 2017.

[19] J. M. M. Montiel, Javier Civera, and Andrew J. Davison. Unified Inverse Depth Parametrization for Monocular SLAM. *Proceedings of Robotics Science & Systems*, 24(5):16–19, 2006.

[20] Martin P. Parsley and Simon J. Julier. Avoiding negative depth in inverse depth bearing-only SLAM. *2008 IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS*, (1):2066–2071, 2008.