Inertial Velocity and Attitude Estimation for Quadrotors

James Svacha[1], Kartik Mohta[1], Michael Watterson[1], Giuseppe Loianno[2], and Vijay Kumar[1]

***Abstract*— This work addresses the design and implementation of a filter that estimates the orientation of the body-fixed z axis and the velocity of a quadrotor UAV from the inertial measurement unit (IMU) given a known yaw. The velocity and attitude estimation is possible since the filter employs a linear drag model measuring the drag forces on the quadrotor through the IMU. These forces are functions of the robot's velocity and attitude. In addition, the filter estimates the linear drag parameters and thrust coefficient for the propellers. These parameters may be fed back into a controller to improve tracking performance. Experimental results are used to validate the proposed approach.**

I. Introduction

Micro Aerial Vehicles (MAVs) equipped with onboard sensors can help humans in a variety of different applications such as search and rescue [1], manipulation [2], [3], transportation [4], exploration [5], inspection [6], and monitoring [7]. Onboard exteroceptive sensors such as LIDAR [8] and cameras are generally used to estimate the state of the vehicle for navigation purposes. Specifically, cameras are useful sensors to estimate the velocity of the platforms using methods such as optical flow [9] and visual-inertial odometry [10]. However, these sensors may be subject to failure and in many applications it is not possible to use them due to computational and/or payload constraints. Furthermore, certain environments make it challenging or impossible to use them. In these environments, the MAV must rely on other sensors for its odometry, such as an inertial measurement unit (IMU), which contains accelerometers, gyroscopes, and sometimes a magnetometer. IMUs are now available on most quadrotor platforms.

This work describes the algorithm design and implementation of an attitude and velocity estimator for quadrotors that only requires an IMU and a known yaw angle. This filter incorporates a linear drag model to concurrently estimate the velocity, aerodynamic model parameters, and accelerometer biases. We outline a Riemannian unscented Kalman filter (UKF) that estimates the tilt component of the attitude on the 2-sphere. Riemannian UKFs are described generally in [11].

Fig. 1: The Ascending Technologies Hummingbird used in our experiments.

Previous works have addressed the use of accelerometers for state estimation and control of quadrotors. In [12], authors developed a detailed model for a quadrotor that includes aerodynamic effects and used the estimated values for feedback control of a quadrotor. Although the model introduced in this work is simpler than that proposed in [12], it still captures the predominant aerodynamic effects on the quadrotor at low to moderate speeds. The authors in [13] incorporated the IMU measurements into an extended Kalman filter (EKF) framework to estimate the quadrotor's roll and pitch and the velocity projected onto the horizontal plane of the body-fixed frame of the quadrotor. The authors in [14] reach this result, and they estimate the drag coefficient as well. In the previous works, the vertical velocity in the body frame cannot be estimated because there is not an explicit model relating the accelerometer measurement directly to this velocity. This work addresses the issue by using a simple linear drag model that has been validated experimentally in [15]. In [16], the authors developed a nonlinear observer to estimate the attitude and the full velocity in the body-fixed frame. However, their results

This work was supported by the ARL grants W911NF-08-2-0004, W911NF-17-2-0181, ONR grants N00014-07-1-0829, N00014-14-1-0510, ARO grant W911NF-13-1-0350, NSF grants IIS-1426840, IIS-1138847, DARPA grants HR001151626, HR0011516850.

[1]The authors are with the GRASP Lab, University of Pennsylvania, 3330 Walnut Street, Philadelphia, PA 19104, USA. email: {jsvacha, kmohta, wami, kumar}@seas.upenn.edu.

[2]The author is with the New York University, Tandon School of Engineering, 6 MetroTech Center, 11201 Brooklyn NY, USA. email: {loiannog}@nyu.edu.

are only obtained through simulations, they assume the power supplied to the motors is known, and they do not estimate the aerodynamic model parameters (e.g., thrust and drag coefficients) in a way such that they can be fed back for control purposes.

The contributions of this paper are twofold. First, we develop a filter to estimate the attitude and 3D velocity of the quadrotor using only the IMU and a known yaw angle. The yaw can generally be obtained using magnetometers or integrating angular rates for short term estimation. However, these methods are prone to errors and drift. For example, it is well known that magnetometers may suffer in proximity to high current power lines and due to magnetic fields generated by electric motors. Because of these problems, we provide a sensitivity analysis quantifying the effects of using an erroneous yaw value in the estimator. The results show that the estimator provides a reasonable output when dead-reckoning the yaw from the IMU, so the filter can be used as a short-term fallback solution in the event that the vision system fails (e.g., during a transition in lighting). Moreover, the proposed filter formulation does not prevent the incorporation of additional sources of more precise and accurate information from cameras and lasers. Second, we show a way to estimate the parameters for the aerodynamic model proposed in [15], [17] online. These parameters are propeller-dependent and have been demonstrated to significantly change over time (due to propeller damage, for instance), so it is useful to have a scheme that can re-estimate them once they change without having to run tedious experiments for system identification. We believe this filter solution can provide a useful navigation solution for nano platforms due to the size, computation, and payload constraints and on larger scale vehicles in case of additional sensor failures.

The paper outline is as follows. Section II introduces preliminary notations and the Hopf fibration. Section III presents an overview of the quadrotor's aerodynamic model. Section IV introduces an Unscented Kalman Filter for the attitude, velocity, aerodynamic model parameters and accelerometer biases. Section V provides experimental results for the filter and shows the effects of a wrong yaw in the estimator through a sensitivity analysis. Section VI concludes the paper and presents future scenarios. A supplementary material [18] derives a result for computing the parallel transport of a vector on S^2, as well as pseudocode for the algorithms.

II. Preliminaries

We model the quadrotor using an inertial frame $\mathcal{W}$ attached to the world which has the basis vectors $\mathbf{e}_1$ pointing North, $\mathbf{e}_2$ pointing West, and $\mathbf{e}_3$ pointing upwards. Rigidly attached to the quadrotor is a coordinate frame $\mathcal{B}$, called the body-fixed frame, with basis vectors $\mathbf{b}_1$ pointing in the quadrotor's frontal direction, $\mathbf{b}_2$ pointing to the left of the quadrotor, and $\mathbf{b}_3$ pointing in the direction of generated thrust. An illustration of the frames $\mathcal{W}$ and $\mathcal{B}$ is shown in Figure 2.

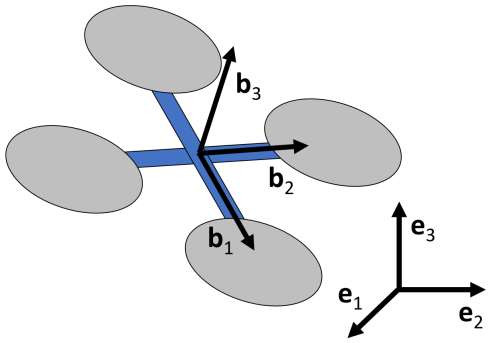

Fig. 2: The body-fixed frame of a quadrotor and the fixed world frame.

The orientation of the quadrotor is represented by the rotation matrix $R = \begin{bmatrix}\mathbf{b}_1 & \mathbf{b}_2 & \mathbf{b}_3\end{bmatrix} \in SO(3)$. Orientations not corresponding to a 180° inversion (the case when the quadrotor is upside-down) can be decomposed into a tilt of the $\mathbf{b}_3$ axis, which is represented by a point on the two dimensional sphere S^2, followed by a yaw about that axis, which is represented by a point on the one-dimensional sphere S^1. We will focus on the estimation of tilt and assume that yaw can be obtained independently from the IMU. This decomposition can be described by the Hopf fibration, and is shown in Figure 3. Given a vector $\mathbf{b}_3 = \begin{bmatrix}b_{3x} & b_{3y} & b_{3z}\end{bmatrix}^\top \in S^2$ and a yaw angle ψ, then, assuming $b_{3z} \neq -1$, the rotation matrix R can be computed as

$$R = R_{\text{tilt}}(\mathbf{b}_3) R_{\text{yaw}}(\psi), \tag{1}$$

where

$$R_{\text{tilt}}(\mathbf{b}_3) = \begin{bmatrix} b_{3z} + \frac{b_{3y}^2}{1+b_{3z}} & -\frac{b_{3x}b_{3y}}{1+b_{3z}} & b_{3x} \\ -\frac{b_{3x}b_{3y}}{1+b_{3z}} & 1 - \frac{b_{3y}^2}{1+b_{3z}} & b_{3y} \\ -b_{3x} & -b_{3y} & b_{3z} \end{bmatrix}, \tag{2}$$

$$R_{\text{yaw}}(\psi) = \begin{bmatrix} \cos\psi & -\sin\psi & 0 \\ \sin\psi & \cos\psi & 0 \\ 0 & 0 & 1 \end{bmatrix}. \tag{3}$$

III. Modeling

We now describe the first-order drag aerodynamic model of a quadrotor, adopted from [15]. The forces acting on the quadrotor are shown in Figure 4. The induced drag force $\mathbf{D}$ is given:

$$\mathbf{D} = -k_d \omega_s R P^\top \mathbf{v}_b, \tag{4}$$

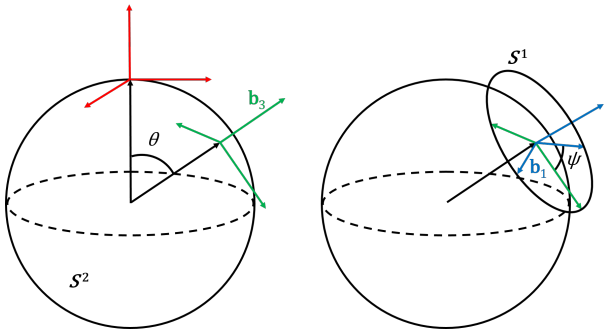

Fig. 3: Almost all orientations in $SO(3)$ may be decomposed into a tilt from the red frame into the green frame, followed by a yaw from the green frame to the blue frame.

where k_d is the drag proportionality constant, $\omega_s = \sum_{i=1}^{4} \omega_i$ is the sum of the motor speeds, $\mathbf{v}_b = \begin{bmatrix} v_{bx} & v_{by} & v_{bz} \end{bmatrix}^\top$ is the velocity of the quadrotor expressed in the basis of the body-fixed frame, and $P = \text{diag}(\begin{bmatrix} 1 & 1 & 0 \end{bmatrix})$ is a matrix that projects a vector onto the x-y plane. The model for the thrust F is also given by [15]

$$F = k_\omega \omega_{ss} - k_z \omega_s v_{bz} + k_h v_{bh}^2, \tag{5}$$

where $\omega_{ss} = \sum_{i=1}^{4} \omega_i^2$ is the sum of the squared motor speeds, k_ω, k_z and k_h are positive proportionality constants and $v_{bh} = \sqrt{v_{bx}^2 + v_{by}^2}$. However, k_h is insignificant at low to moderate speeds for our platform, so we simplify the thrust model by removing the k_h term

$$F = k_\omega \omega_{ss} - k_z \omega_s v_{bz}. \tag{6}$$

Thus, we have the model for the acceleration $\mathbf{a}$ of the quadrotor

$$\mathbf{a} = \frac{1}{m} k_\omega \omega_{ss} \mathbf{b}_3 - \frac{1}{m} \omega_s R D \mathbf{v}_b - g \mathbf{e}_3, \tag{7}$$

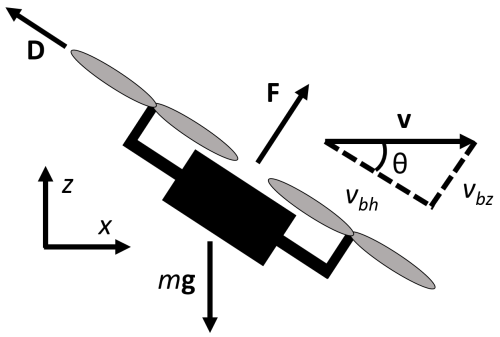

Fig. 4: The forces acting on the quadrotor, as well as the velocity and its components in the body-fixed frame.

where m is the mass of the robot, g is the magnitude of the gravity vector $\mathbf{g}$ and $D = \text{diag}(\begin{bmatrix} k_d & k_d & k_z \end{bmatrix})$ is a diagonal matrix containing the horizontal and vertical drag coefficients.

The acceleration of the quadrotor in the body-fixed frame, $\mathbf{a}_b$, is given by

$$\mathbf{a}_b = \frac{1}{m} k_\omega \omega_{ss} \mathbf{e}_3 - \frac{1}{m} \omega_s D \mathbf{v}_b - g R^\top \mathbf{e}_3, \tag{8}$$

where we have, for simplicity of notation, dropped the dependence of the drag and thrust parameters on t. Furthermore, we have the equation

$$\mathbf{v} = R \mathbf{v}_b. \tag{9}$$

Taking the time derivative and left-multiplying by $R^\top$ provides

$$\mathbf{a}_b = R^\top \dot{R} \mathbf{v}_b + \dot{\mathbf{v}}_b. \tag{10}$$

Furthermore, using the fact that $R^\top \dot{R} \mathbf{v}_b = \boldsymbol{\Omega}_b \times \mathbf{v}_b$, where $\boldsymbol{\Omega}_b$ is the angular velocity of the body-fixed frame, expressed in the basis of the body-fixed frame, we have

$$\dot{\mathbf{v}}_b = \mathbf{a}_b - \boldsymbol{\Omega}_b \times \mathbf{v}_b. \tag{11}$$

Hence, we have

$$\dot{\mathbf{v}}_b = \frac{1}{m} k_\omega \omega_{ss} \mathbf{e}_3 - \frac{1}{m} \omega_s D \mathbf{v}_b - g R^\top \mathbf{e}_3 - \boldsymbol{\Omega}_b \times \mathbf{v}_b. \tag{12}$$

The attitude "tilt" is represented by stereographic coordinates (s_x, s_y) on S^2. Given a point $\mathbf{b}_3 = \begin{bmatrix} b_{3x} & b_{3y} & b_{3z} \end{bmatrix}^\top \in S^2$ such that $\|\mathbf{b}_3\| = 1$ and $b_{3z} \neq -1$, the stereographic coordinates for that point are obtained by projecting it onto the x-y plane through the point $(0, 0, -1)$, and are given

$$s_x = \frac{b_{3x}}{1 + b_{3z}}, \quad s_y = \frac{b_{3y}}{1 + b_{3z}}, \tag{13}$$

and the mapping from stereographic coordinates to $\mathbf{b}_3$ is given

$$\begin{aligned} b_{3x} &= \frac{2 s_x}{1 + s_x^2 + s_y^2}, \\ b_{3y} &= \frac{2 s_y}{1 + s_x^2 + s_y^2}, \\ b_{3z} &= \frac{1 - s_x^2 - s_y^2}{1 + s_x^2 + s_y^2}. \end{aligned} \tag{14}$$

The derivatives for these coordinates as a function of angular velocity in the body-fixed frame $\boldsymbol{\Omega}_b = \begin{bmatrix} p & q & r \end{bmatrix}^\top$ and the yaw angle can be obtained by solving the equation:

$$\frac{d}{dt} \mathbf{b}_3 = (R \boldsymbol{\Omega}_b) \times \mathbf{b}_3, \tag{15}$$

for $\dot{s}_x$ and $\dot{s}_y$ after substituting eq. (14) for $\mathbf{b}_3$ and eq. (1) for R. The result is:

$$\dot{s}_x = \frac{1}{2}(q\cos\psi + p\sin\psi)(1 + s_x^2 + s_y^2), \tag{16}$$

$$\dot{s}_y = \frac{1}{2}(q\sin\psi - p\cos\psi)(1 + s_x^2 + s_y^2). \tag{17}$$

IV. State Estimation

This section describes the outline of an Unscented Kalman Filter to estimate the velocity and orientation on S^2. We denote the state $\mathbf{x}$ of the filter by

$$\mathbf{x} = \begin{bmatrix}\mathbf{s}^\top & \mathbf{v}_b^\top & \mathbf{k}^\top & \mathbf{b}_a^\top\end{bmatrix}^\top, \tag{18}$$

where $\mathbf{s} = \begin{bmatrix}s_x & s_y\end{bmatrix}^\top$ is a vector of the stereographic coordinates of the vector $\mathbf{b}_3$, $\mathbf{k} = \begin{bmatrix}k_\omega & k_d & k_z\end{bmatrix}^\top$ is a vector of the drag coefficients, and $\mathbf{b}_a = \begin{bmatrix}b_{ax} & b_{ay} & b_{az}\end{bmatrix}^\top$ is the vector of accelerometer biases. For convenience, we denote the partition of the state vector that does not include $\mathbf{s}$ by $\mathbf{s}' = \begin{bmatrix}\mathbf{v}^\top & \mathbf{k}^\top & \mathbf{b}_a^\top\end{bmatrix}^\top$. We denote by $\hat{[\cdot]}$ an estimate of the quantity $[\cdot]$, such as the state estimate $\hat{\mathbf{x}}$ or the estimate of the error covariance $\hat{P}$. Furthermore, we denote the time of the estimate with a subscript, such that at the kth time step t_k, we have the state estimate $\hat{\mathbf{x}}_k$ and the error covariance estimate $\hat{P}_k$.

In order to estimate $\mathbf{s}$ using an UKF, we must know how to compute the sigma points corresponding to the covariance on the 2-sphere, S^2. The covariance of $\mathbf{s}$ is only defined on the tangent space of the 2-sphere, and its components are numerically determined by a tangent basis. As the estimate of $\mathbf{s}$ changes, we must parallel transport this tangent basis on the manifold so that we can compute the sigma points at each time step.

Fortunately, the parallel transport on S^2 using the Levi-Civita connection corresponding to the metric induced by $\mathbb{R}^3$ is intuitive. In the supplementary material [18], we show that parallel transporting a vector $\mathbf{v}_p \in T_pS^2$ to $\mathbf{v}_q \in T_qS^2$ along the geodesic from the point $p \in S^2$ to the point $q \in S^2$ is the same as rotating the vector $\mathbf{v}_p$ by the rotation matrix R_{qp} that directly rotates the vector $\mathbf{p} \in \mathbb{R}^3$ representing the point p into the vector $\mathbf{q} \in \mathbb{R}^3$ representing the point q, that is

$$\mathbf{v}_q = R_{qp}\mathbf{v}_p,\ R_{qp} = \exp(\theta_{pq}[\boldsymbol{\omega}]_\times), \tag{19}$$

where $\boldsymbol{\omega} \in \mathbb{R}^3$ is orthogonal to both $\mathbf{p}$ and $\mathbf{q}$, and $\theta_{pq} \in \mathbb{R}$ is the angle of rotation from p to q. Furthermore, $[\boldsymbol{\omega}]_\times \in \mathbb{R}^{3\times 3}$ represents the skew-symmetric matrix such that, for any vector $\mathbf{v} \in \mathbb{R}^3$, $\boldsymbol{\omega}\times\mathbf{v} = [\boldsymbol{\omega}]_\times\mathbf{v}$.

We now describe the filtering algorithm. The current tangent basis vectors at the kth time step t_k, which are obtained through parallel transport, are denoted $\mathbf{t}_{xk}$ and $\mathbf{t}_{yk}$. Together, these vectors form the matrix $T_k = \begin{bmatrix}\mathbf{t}_{xk} & \mathbf{t}_{yk}\end{bmatrix}$. When we compute the sigma points, we do a Cholesky factorization of the covariance and then express the sigma points using the current tangent basis. So, if the covariance at time k is $\hat{P}_k$, then we compute $L_k = \sqrt{(n+\lambda)\hat{P}_k}$, where n is the dimension of the state space and λ is a tunable parameter. For now, we suppress the subscript k to simplify notation. L becomes

$$L = \begin{bmatrix}\boldsymbol{\delta}_1 & \boldsymbol{\delta}_2 & \cdots & \boldsymbol{\delta}_n \\ \boldsymbol{\delta}'_1 & \boldsymbol{\delta}'_2 & \cdots & \boldsymbol{\delta}'_n\end{bmatrix}, \tag{20}$$

where $\boldsymbol{\delta}_i$ represents the deviation of the ith sigma point from the current rotation ($\hat{\mathbf{s}}_k$) on the tangent space of the sphere, and $\boldsymbol{\delta}'_i$ represents the deviation of the ith sigma point from the remainder of the state vector ($\hat{\mathbf{s}}'_k$). The first sigma point is $\mathcal{X}_0 = \hat{\mathbf{x}}$, and the remaining $2n$ sigma points are represented, for $i = 1, \ldots, n$

$$\mathcal{X}_i = \begin{bmatrix}\exp_{\hat{\mathbf{s}}}(T\boldsymbol{\delta}_i) \\ \mathbf{s}' + \boldsymbol{\delta}'_i\end{bmatrix}, \quad \mathcal{X}_{n+i} = \begin{bmatrix}\exp_{\hat{\mathbf{s}}}(-T\boldsymbol{\delta}_i) \\ \mathbf{s}' - \boldsymbol{\delta}'_i\end{bmatrix}, \tag{21}$$

where $\exp_{\hat{\mathbf{s}}}(\cdot)$ maps a vector in the tangent space of $\hat{\mathbf{s}}$ on the sphere to the manifold itself as follows. Given a point $p \in S^2$ and a vector $\mathbf{v} \in T_p(S^2)$, the exponential map finds the point $q = \exp_p(\mathbf{v})$ computed by traveling a distance $\|\mathbf{v}\|$ along the geodesic in the direction of $\mathbf{v}$. This approach of performing computations directly on the manifold is particularly useful for larger time steps and larger values of λ. If the vector expression for the point p is $\mathbf{p}$ and that of q is $\mathbf{q}$, then we can construct a rotation matrix R_{pq} to get $\mathbf{q} = R_{pq}\mathbf{p}$. R_{pq} is constructed using Rodrigues' formula, with the axis $\mathbf{u}$ and angle θ

$$\mathbf{u} = \frac{\mathbf{p}\times\mathbf{v}}{\|\mathbf{p}\times\mathbf{v}\|}, \quad \theta = \|\mathbf{v}\|. \tag{22}$$

The sigma points at the previous time step t_{k-1}, $\mathcal{X}_{i,k-1}$, are transformed into new sigma points $\mathcal{X}^-_{i,k}$ using the nonlinear dynamics of the system, $\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, \mathbf{u})$. In these dynamics, we consider $\mathbf{k}$ as a constant and $\mathbf{b}_a$ as a random walk.

$$\dot{\mathbf{k}} = 0, \quad \dot{\mathbf{b}}_a = \boldsymbol{\eta}_{ba}, \tag{23}$$

where $\boldsymbol{\eta}_{ba}$ is Gaussian noise. The rest of the state transforms according to the dynamics given by equations (8) and (17).

The propagated sigma points $\mathcal{X}_i^-$ may be partitioned again into the rotation components $\mathcal{S}_i$ and the remainder of the state, $\mathcal{S}'_i$, so that $\mathcal{X}_i^- = \begin{bmatrix}\mathcal{S}_i^\top & \mathcal{S}_i'^\top\end{bmatrix}^\top$.

The weighted mean of the $\mathcal{S}_i$'s is computed on the sphere to get $\hat{\mathbf{s}}^-$ using the algorithm WeightedAvgSphere, which is described in the supplementary material and is generalized for a larger class of manifolds in [19].

The mean of the $\mathcal{S}_i'$'s are computed by a weighted average to get $\hat{\mathbf{s}}'^{-}$. After averaging, we have the a-priori state estimate $\hat{\mathbf{x}}^- = \begin{bmatrix}\hat{\mathbf{s}}^{-\top} & \hat{\mathbf{s}}'^{-\top}\end{bmatrix}^\top$. After updating the state, the orientation will lie at a different point on the sphere. Since the covariance is expressed with respect to the previous mean, we need to parallel transport the tangent basis T_{k-1} along the sphere to get a new tangent basis, T_k^-, in which we will express the new sigma point deviations. The deviations of the transformed sigma points are recomputed to calculate the covariance $\hat{P}^-$ after the process update. For the orientation component of the sigma points, this is done by lifting the sigma points off of the manifold and into the tangent space using the log map, which is the inverse of the exponential map on the 2-sphere. We express the new tangent vectors computed using the log map in the basis T^-. This is done by taking the dot product of each basis vector in T^- with the logarithms of the $\mathcal{S}_i$'s

$$\boldsymbol{\delta}_i^- = T^{-\top} \log_{\hat{\mathbf{s}}^-}(\mathcal{S}_i). \tag{24}$$

The log map is computed as follows: given two points p and q on S^2, with vector representations $\mathbf{p}$ and $\mathbf{q}$, we wish to find the vector $\mathbf{v} = \log_p(q) \in T_p(S^2)$ whose direction and magnitude represent the direction and length of the geodesic from p to q. Assuming that $\|\mathbf{v}\| < \pi$, then we can compute the direction $\hat{\mathbf{v}}$

$$\hat{\mathbf{v}} = \frac{(\mathbf{p}\times\mathbf{q})\times\mathbf{p}}{\|(\mathbf{p}\times\mathbf{q})\times\mathbf{p}\|},\ \|\mathbf{v}\| = \cos^{-1}(\mathbf{p}\cdot\mathbf{q}). \tag{25}$$

After computing the deviations $\boldsymbol{\delta}_i^-$, the deviations of the remainder of the state vector $\boldsymbol{\delta}_i'^-$ are just computed by taking the difference $\mathcal{S}_i' - \hat{\mathbf{s}}'^-$. The covariance is recomputed by taking the weighted outer product of the deviations, and adding the process noise covariance matrix Q

$$\hat{P}^- = \sum_{i=0}^{n} w_{ci} \begin{bmatrix}\boldsymbol{\delta}_i^- \\ \boldsymbol{\delta}_i'^-\end{bmatrix} \begin{bmatrix}\boldsymbol{\delta}_i^{-\top} & \boldsymbol{\delta}_i'^{-\top}\end{bmatrix} + Q. \tag{26}$$

We transform the sigma points $\mathcal{X}_i^-$ into the measurement space to generate new sigma points $\mathcal{Y}_i$

$$\mathcal{Y}_i = \mathbf{h}(\mathcal{X}_i^-), \tag{27}$$

where $\mathbf{h}$ is the measurement transformation describing the accelerometer measurement as a function of the state

$$\mathbf{h}(\hat{\mathbf{x}}) = \frac{k_\omega}{m}\omega_{ss}\mathbf{e}_3 - \frac{\omega_s}{m}D\mathbf{v}_b + \mathbf{b}_a. \tag{28}$$

We may then compute the mean $\hat{\mathbf{y}}$ and covariance $\hat{P}_{yy}$ of the sigma points $\mathcal{Y}_i$, the cross covariance $\hat{P}_{xy}$, and the Kalman gain K. Then, we can compute the state correction, $\Delta_x = K(\mathbf{y} - \hat{\mathbf{y}})$. This correction is partitioned into an orientation correction Δ_s and a correction on the remainder of the state $\Delta_{s'}$. The correction Δ_s lies in the tangent space of S^2 at $\hat{\mathbf{s}}^-$, so we compute the new orientation $\hat{\mathbf{s}} = \exp_{\hat{\mathbf{s}}^-}(T^-\Delta_s)$. The remainder of the state vector is updated by adding $\Delta_{s'}$ to $\hat{\mathbf{s}}'^-$. Finally, we update the covariance and parallel transport the tangent basis T^- to the new tangent basis T to account for the measurement update.

V. Experimental Results

In this section, we report on the experiments performed at the PERCH lab (Penn Engineering Research Collaborative Hub) at the University of Pennsylvania indoor testbed. The total flying area volume is $20 \times 6 \times 4$ m^3. A Vicon [1] motion capture system with 20 cameras running at 100 Hz is used for ground-truth. Experiments were performed using data collected on an Ascending Technologies Hummingbird outfitted with an Odroid XU-4, where the high-level control was performed. The yaw is obtained from the motion capture system.

We now list the UKF parameters. The state is initialized to the zero vector. All covariances are diagonal. The initial tilt and velocity covariance entries are all 0.01, and the units for the tilt are radians and those for velocity are m/s. The initial covariance entries for $\mathbf{k}$ are $\begin{bmatrix}1.0\text{e-}08 & 1.0\text{e-}12 & 1.0\text{e-}08\end{bmatrix}$. k_d and k_z have units of $\frac{\text{kg}\cdot\text{N}}{\text{RPM}\cdot\text{m/s}}$, while k_ω has units of $\frac{\text{kg}\cdot\text{N}}{\text{RPM}^2}$. The initial covariance entries for $\mathbf{b}_a$ are 0.1, and the units for bias are m/s^2. The entries of the process noise covariance are 1.0e-03 for the tilt, 0.1 for the velocity, $\begin{bmatrix}1.0\text{e-}14 & 1.0\text{e-}20 & 1.0\text{e-}14\end{bmatrix}$ for $\mathbf{k}$, and 1.0e-03 for the accelerometer bias. Finally, the measurement covariance for the accelerometer is $\begin{bmatrix}0.2 & 0.2 & 0.5\end{bmatrix}$. The UKF parameters α, κ, and β are 0.1, 0, and -2, respectively.

A. Trajectory Trials

Figures 5 and 6 overlay the estimated attitude and velocity, respectively, for 14 trials of a Lissajous trajectory. The RMS and standard deviations of the error are shown for the filter in Table I. Figure 7 shows the velocity errors for a 520 second run of the filter on the Lissajous trajectory. The Lissajous has the same form as in Figure 6, but over a longer duration. Figure 8 shows the estimated aerodynamic model parameters for the same experiment. Note that the k_ω estimate decays over time. This is because the RPM output reading from the vehicle depends on battery voltage. Thus, although $\dot{\mathbf{k}} = 0$ in theory, in practice some process noise can be added to compensate for these effects. The performance of the algorithm is also evaluated considering the CPU

[1] http://www.vicon.com/

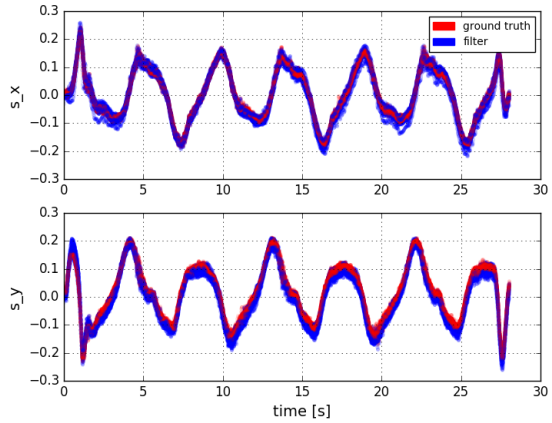

Fig. 5: S^2 attitude components estimated by our filter compared to ground truth during 14 repeated autonomous flight trials.

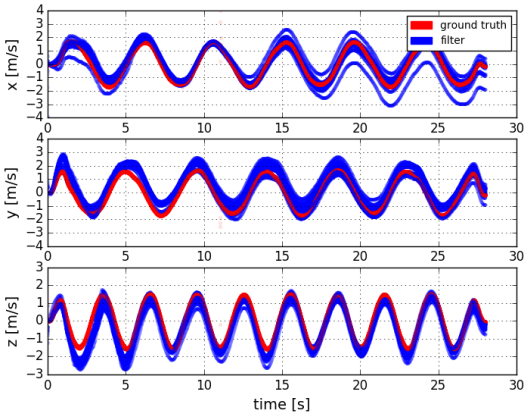

Fig. 6: Velocity estimated by our filter compared to ground truth during repeated trials of autonomous flight.

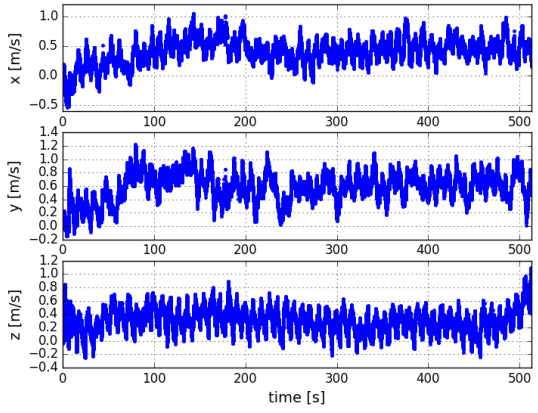

Fig. 7: Velocity errors for the Lissajous trajectory during a 520 s trial.

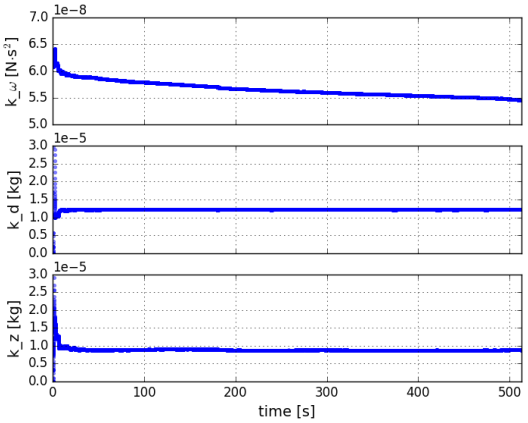

Fig. 8: Aerodynamic model parameters for the 520 s run of the Lissajous trajectory.

time on a i7 laptop and an Odroid XU-4 as shown in Table II. The algorithm is suitable for computational limited CPUs.

	vx	vy	vz	v total	sx	sy	s total
RMS	0.417	0.618	0.432	0.862	0.0118	0.0166	0.0204
STD	0.406	0.366	0.312	N/A	0.0112	0.0124	N/A

TABLE I: RMS and standard deviation of velocity and attitude errors during autonomous flight.

CPU time (ms)	Laptop i5	Odroid XU-4
prediction	0.111	0.219
update	0.0223	0.0488

TABLE II: Performance comparison on different CPUs.

B. Sensitivity Analysis

As previously mentioned, the estimator requires a known yaw angle as defined in Section II. However, the yaw is subject to drifts and errors when estimated using dead reckoning from the IMU or magnetometers. To quantify the effect of a wrong yaw estimate on the stereographic coordinates, we propose a sensitivity analysis based on the yaw estimated from the IMU and evaluate the result with respect to ground truth. We perform the analysis by linearizing equation (17) and evaluating the Jacobians at the current state

$$\dot{\delta s} = J_s \delta \mathbf{s} + J_\psi \delta\psi, \tag{29}$$

where $J_s = \left(\frac{\partial}{\partial \mathbf{s}}\dot{\mathbf{s}}\right)|_{\mathbf{s}=\mathbf{s}^*,\psi=\psi^*}$, $J_\psi = \left(\frac{\partial}{\partial \psi}\dot{\mathbf{s}}\right)|_{\mathbf{s}=\mathbf{s}^*,\psi=\psi^*}$, $\mathbf{s}^*$ is the current tilt, ψ^* is the ground truth yaw, $\delta\mathbf{s} = \mathbf{s} - \mathbf{s}^*$ is the error in the estimate of $\mathbf{s}$, and $\delta\psi = \psi - \psi^*$ is the error in the dead-reckoned yaw ψ output by the IMU with the yaw ψ^* obtained from Vicon. To compute $\delta\mathbf{s}$ in eq. (29), we discretize the system using a first order Euler approximation. At each time step,

we evaluate the Jacobians at $\mathbf{s}^*$. The resulting plot of stereographic coordinate errors and the corresponding tilt angle error (the angle between the two points on S^2) is presented in Figure 9 for the first minute of the 520 second-long test in Figure 7. The analysis shows that the results are similar to the estimated ones. Within the first 40 seconds, the maximum tilt angle error is around 5 degrees, showing the robustness to yaw errors.

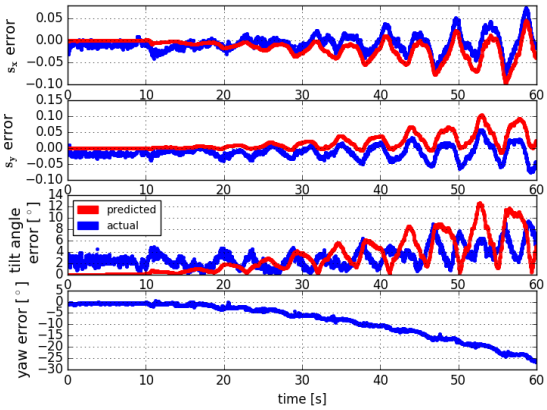

Fig. 9: Predicted tilt error (red) using the yaw obtained from the IMU, compared with the actual tilt error (blue).

VI. Conclusion

In this work, a novel velocity and attitude estimator was developed and its performance was demonstrated during repeated autonomous flights. The proposed solution can provide benefits to the autonomy of nano scale platforms due to their limited sensor payload and lower robustness against sensor failures than larger vehicles.

Future works involve comparing this solution to other nonlinear observers. We are also interested to extend the filter by adding angular velocity to the state vector and the differential motor speeds (roll, pitch, and yaw moments) as inputs to the system. This will allow for estimation of the inertia tensor as well.

References

[1] N. Michael, S. Shen, K. Mohta, Y. Mulgaonkar, V. Kumar, K. Nagatani, Y. Okada, S. Kiribayashi, K. Otake, K. Yoshida, K. Ohno, E. Takeuchi, and S. Tadokoro, "Collaborative mapping of an earthquake-damaged building via ground and aerial robots," *Journal of Field Robotics*, vol. 29, no. 5, pp. 832–841, 2012.

[2] F. Forte, R. Naldi, and L. Marconi, "Impedance Control of an Aerial Manipulator," in *American Control Conference (ACC)*, Montreal, Canada, 2012, pp. 3839–3844.

[3] J. Thomas, G. Loianno, K. Daniilidis, and V. Kumar, "Visual servoing of quadrotors for perching by hanging from cylindrical objects," *IEEE Robotics and Automation Letters*, vol. 1, no. 1, pp. 57–64, Jan 2016.

[4] G. Loianno and V. Kumar, "Cooperative transportation using small quadrotors using monocular vision and inertial sensing," *IEEE Robotics and Automation Letters*, vol. 3, no. 2, pp. 680–687, April 2018.

[5] T. Tomic, K. Schmid, P. Lutz, A. Domel, M. Kassecker, E. Mair, I. Grixa, F. Ruess, M. Suppa, and D. Burschka, "Toward a fully autonomous uav: Research platform for indoor and outdoor urban search and rescue," *IEEE Robotics Automation Magazine*, vol. 19, no. 3, pp. 46–56, Sept 2012.

[6] T. Ozaslan, G. Loianno, J. Keller, C. J. Taylor, V. Kumar, J. M. Wozencraft, and T. Hood, "Autonomous navigation and mapping for inspection of penstocks and tunnels with MAVs," *IEEE Robotics and Automation Letters*, vol. 2, no. 3, pp. 1740–1747, July 2017.

[7] J. Hodgson, S. Baylis, R. Mott, A. Herrod, and R. Clarke, "Precision wildlife monitoring using unmanned aerial vehicles," *Nature - Scientific Reports*, Mar. 2016.

[8] J. Zhang and S. Singh, "Loam: Lidar odometry and mapping in real-time," in *Robotics: Science and Systems Conference*, Pittsburgh, PA, July 2014.

[9] V. Grabe, H. H. Blthoff, and P. R. Giordano, "On-board velocity estimation and closed-loop control of a quadrotor uav based on optical flow," in *Proc. IEEE International Conference on Robotics and Automation (ICRA)*, Saint Paul, Minnesota, May 2012, pp. 491–497.

[10] M. Bloesch, S. Omari, M. Hutter, and R. Siegwart, "Robust visual inertial odometry using a direct EKF-based approach," in *Proc. IEEE International Conference on Intelligent Robots and Systems (IROS)*, Hamburg, Germany, Sep. 2015, pp. 298–304.

[11] S. Hauberg, F. Lauze, and K. S. Pedersen, "Unscented kalman filtering on riemannian manifolds," *Journal of Mathematical Imaging and Vision*, vol. 46, no. 1, pp. 103–120, May 2013.

[12] P. Martin and E. Salaun, "The true role of accelerometer feedback in quadrotor control," in *Proc. IEEE International Conference on Robotics and Automation (ICRA)*, Anchorage, Alaska, May 2010, pp. 1623–1629.

[13] D. Abeywardena, S. Kodagoda, G. Dissanayake, and R. Munasinghe, "Improved state estimation in quadrotor mavs: A novel drift-free velocity estimator," *IEEE Robotics & Automation Magazine*, vol. 20, Jun. 2013.

[14] R. Leishman, J. C. Macdonald, Jr., R. Beard, and T. McLain, "Quadrotors and accelerometers: State estimation with an improved dynamic model," *IEEE Control Systems Magazine*, vol. 34, Feb. 2014.

[15] J. Svacha, K. Mohta, and V. Kumar, "Improving quadrotor trajectory tracking by compensating for aerodynamic effects," in *Proc. IEEE Conference on Unmanned Aircraft Systems (ICUAS)*, Miami, Florida, 2017, pp. 860–866.

[16] G. Allibert, D. Abeywardena, M. Bangura, and R. Mahony, "Estimating body-fixed frame velocity and attitude from inertial measurements for a quadrotor vehicle," in *Proc. Multi-Conference on Systems and Control, address =*.

[17] M. Faessler, A. Franchi, and D. Scaramuzza, "Differential flatness of quadrotor dynamics subject to rotor drag for accurate tracking of high-speed trajectories," *IEEE Robotics and Automation Letters*, vol. 3, no. 2, pp. 620–626, April 2018.

[18] J. Svacha, K. Mohta, M. Watterson, G. Loianno, and V. Kumar, "Inertial velocity and attitude estimation for quadrotors: Supplementary material," Feb. 2018. [Online]. Available: http://repository.upenn.edu/ese_papers/834

[19] C. Hertzberg, R. Wagner, U. Frese, and L. Schrder, "Integrating generic sensor fusion algorithms with sound state representations through encapsulation of manifolds," *Information Fusion*, vol. 41, no. 1, pp. 57–77, 2013.