Measurement model and observability analysis for optical flow-aided inertial navigation

Jia Deng
Sentang Wu
Hongbo Zhao
Da Cai

Measurement model and observability analysis for optical flow-aided inertial navigation

Jia Deng,[a,*] Sentang Wu,[a] Hongbo Zhao,[a] and Da Cai[b]
[a]Beihang University, School of Automation Science and Electrical Engineering, Beijing, China
[b]China Academy of Launch Vehicle Technology, Beijing, China

Abstract. We introduce a navigation filter fused with information of visual optical flow and data collected from an inertial measurement unit during GPS signal degradation. Under the assumption that the tracked feature points are located on a level plane, the feature depth can be explicitly expressed and an exact measurement model was derived. Moreover, an error model analysis for a block-matching-based optical flow algorithm has been investigated. The measurement error follows a Gaussian distribution, which is a prerequisite for leveraging the error-state Kalman filter. Subsequently, a local observability analysis of the proposed filter was performed yielding the expression of three unobservable directions. We emphasize the ability of the proposed filter to estimate the aircraft's state, especially for accurate altitude estimation, without any help of prior knowledge or extra sensors. Finally, an extensive Monte Carlo analysis was used to verify the findings in the observability results showing that all states can be estimated except the absolute horizontal positions and rotation around the gravity vector. The effectiveness of the proposed filter is demonstrated through experimental hardware used to acquire outdoor flight test data.

1 Introduction

During the past decades, there has been a growing research interest in developing vision-based or vision-aided navigation systems for unmanned aerial vehicles (UAVs) under circumstances without prior information.[1–4] The traditional GPS/inertial navigation system (INS) may face severe drift and inaccuracy for the estimation, especially in environments where GPS is not reliable. Integrated inertial systems using sensors such as radar, LiDAR, or sonar can be regarded as a possible method to improve accuracy.[5] However, radar or LiDAR may increase the cost, weight, and power consumption for an airborne platform, while low-cost sonar always shows limited performance for flights in outdoor scenes. Camera-aided navigation and visual odometry, on the other hand, provide rich information about the environment and the camera pose, which seems to be a superior solution to the above problems.

In recent years, simultaneous localization and mapping gathered a lot of attention since it can provide accurate and real-time estimates of six-dimensional parameters. Nevertheless, it is subject to a series of strict requirements such as high-quality images for feature extraction, hardware with powerful onboard processor needed as computational complexity increase related to the quantity of features, and repeated tracking for the same image features over longer periods of time is needed to realize loop-closure.[6] Moreover, there are few applications for fixed-wing aircraft that are faster and higher than quadcopters in an outdoor unstructured environment. Optical flow-aided INS evades the need for mapping and landmark data and shows better tolerance in terms of handling scenes with less salient features and little contrast texture. In addition, it has a compact image representation and does not require the extra burden for image storage without loss of rich navigation information on the translation and rotational velocity as well as the scene depth.[7] It is more suitable for implementation in systems with limited resources. Previous tests have been conducted to compare the performance between different optical flow algorithms for UAV navigation, as introduced in Ref. 8. The optical flow observation error is usually modeled with the default zero mean normal distribution without any explanations.[9] We turn to computer vision scientists for a better answer, but it was found that they mainly focus on the evaluation methodology to give new statistics for indicating the performance.[10] They have raised definitions such as "absolute flow endpoint error" and "frame interpolation error,"[11] rather than proposing a specific model analysis for the flow error. Furthermore, the use of INS aided with optical flow generated by a single camera often adopts an additional ultrasound range sensor to measure the distance to the ground,[12] which solves the ambiguity of the scale factor between the translational velocity magnitude and the scene depth.[9] Another commonly used method assumes that the specific maneuvers for UAV could be designed to keep a constant height or attitude. Five optimization constraints-based methods for the estimation of ego-motion were presented in Florian Raudies' review,[10] which can successively solve the problem. As an example, Thomas P. Webb[13] leveraged epipolar constraint of feature points as the measurement equation in an implicit extended Kalman filter, also termed the essential or coplanarity constraint, and optical flow subspace constraint is similar to its differential version. However, the filter does not perform well in estimating the velocity states as the epipolar constraint weakly relates to velocity. A bilinear constraint-based visual-inertial scheme for relative motion estimation has been introduced.[7] The observability analysis of INS errors for explicit measurements of

*Address all correspondence to Jia Deng, E-mail: dengjia@buaa.edu.cn

a subspace constraint-based optical flow method was investigated in Jacques Waldmann's work.[14] These approaches require sufficient numbers of optical flow features to obtain the corresponding linear equations for the unknowns. Xu et al.[2] presented an autonomous landing method using planar terrain features tracked in sequential images. Using the assumption of a level plane, therefore allowing for the scale of the line-of-sight vector to be expressed explicitly.

Compared to existing work, the main contributions of the proposed research focused on (1) a detailed error model analysis for optical flow measurement based on block-matching algorithm was investigated and showed to obey a Gaussian distribution and therefore suitable for error-state Kalman filter (ESKF); (2) the feature depth was obtained without extra-measurement device through the assumption of flat terrain, which could be reasonably used for outdoor navigation missions when UAVs fly at a high altitude. In the flat terrain assumption, the height proved to be observable by local observability analysis, which was further verified by simulation and experimental results. In addition, a more detailed development of the ESKF is proposed, which fused the information from IMU and optical flow originally described in Refs. 15 and 16. The paper is organized as follows: Sec. 2 describes the problem formulation, introduces the adopted sensors' dynamical model, reviews the computation of an optical flow block-matching algorithm, and derives the measurement error analysis. Section 3 outlines the structure of the ESKF to estimate the full states. Section 4 presents the observability analysis. Sections 5 and 6 show the simulation, experimental results, and corresponding discussion; the final conclusions are given in Sec. 7.

2 Problem Formulation

The geometry of the general case of an aircraft equipped with an IMU and a monocular camera is shown in Fig. 1. Symbol $\{B\}$ represents the aircraft body-fixed frame, and the vehicle relative position to the global inertial reference frame $\{G\}$ is denoted by the vector $^G\mathbf{r}$. The camera frame $\{C\}$ is fixed to the aircraft with a mounting displacement $^B\mathbf{\Delta}$. We denote the coordinates of the j'th feature point P_j in the frame $\{G\}$ by $^G\mathbf{P}_j$, while $^C\mathbf{P}_j$ denotes the relative position of P_j w.r.t.
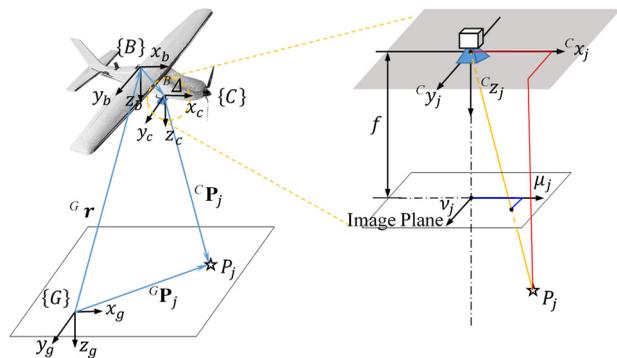
{G}$x_g y_g z_g$---the global reference frame, $^G\mathbf{P}_j$ denotes the coordinates of the feature point P_j in frame $\{G\}$, $^G\mathbf{r}$ denotes the aircraft relative position in frame $\{G\}$.
{B}$x_b y_b z_b$---the aircraft body-fixed frame, $^B\Delta$ denotes the camera's mounting displacement.
{C}$x_c y_c z_c$---the camera frame, $^C\mathbf{P}_j$ denotes the coordinates of the feature point P_j in frame $\{G\}$, i.e. $^C\mathbf{P}_j=[^C x_j, {}^C y_j, {}^C z_j]^\top$.
μ_j ν_j---the projection of the feature P_j on the image plane.
f---the focal length.

Fig. 1 Problem geometry and pinhole camera model.

the camera frame $\{C\}$. The transformation matrices $_B T_G$ and $_C T_B$ describe the rotations from $\{G\}$ to $\{B\}$ and $\{B\}$ to $\{C\}$, respectively, where $_C T_B$ is dependent on the mounting of the camera to the aircraft. The relative position of P_j can be expressed in the camera frame $\{C\}$ as

$$^C\mathbf{P}_j = {}_C T_B {}_B T_G(^G\mathbf{P}_j - {}^G\mathbf{r}) - {}_C T_B {}^B\mathbf{\Delta}. \tag{1}$$

Considering the differential equation with respect to the reference frame $\{G\}$ would require calculating the derivative of the rotation transformation matrix. Therefore, the rotational velocities of the vehicle expressed in the frame $\{B\}$ and $\{C\}$ are represented by $^B\mathbf{\omega}$ and $^C\mathbf{\omega}$, respectively, resulting as

$$^C\dot{\mathbf{P}}_j = {}_C T_B {}_B T_G(^G\dot{\mathbf{P}}_j - {}^G\dot{\mathbf{r}}) - {}_C T_B(^B\dot{\mathbf{\Delta}} + {}^B\mathbf{\omega} \times {}^B\Delta)$$
$$- {}_C T_B \cdot {}^B\mathbf{\omega} \times {}^C\mathbf{P}_j, \tag{2}$$

where $^G\dot{\mathbf{P}}_j = 0$ derives from the assumption for static feature point. Consider a strapdown camera that can be assumed to have a fixed location coinciding with the center of gravity of the aircraft, so that $\{B\}$ and $\{C\}$ can be regarded as the same frame, i.e., $^B\mathbf{\Delta} = {}^B\dot{\mathbf{\Delta}} = 0$, and $_C T_B$ is the identity matrix, which yields the simplified Eq. (3):

$$^C\dot{\mathbf{P}}_j = -{}_B T_G {}^G\dot{\mathbf{r}} - {}^B\mathbf{\omega} \times {}^C\mathbf{P}_j. \tag{3}$$

2.1 Pinhole Camera Model

This work used a pinhole camera model, whose details are shown in Fig. 1. The camera coordinate frame is defined by $^C z$ paralleling to the optical axis, $^C x$ and $^C y$ matching the horizontal and vertical image directions. The projection results of the feature P_j on the image plane are expressed as

$$\begin{bmatrix} \mu_j \\ \nu_j \end{bmatrix} = \frac{f}{^C z_j} \begin{bmatrix} ^C x_j \\ ^C y_j \end{bmatrix}, \tag{4}$$

where f is the focal length, $[\mu_j, \nu_j]^T$ is the projected pixel coordinates of the feature point, and $^C\mathbf{P}_j = [^C x_j, {}^C y_j, {}^C z_j]^T$. With the previous assumption on the identity of the frame $\{B\}$ and $\{C\}$, the coordinate of the feature expressed in the camera frame is given as

$$^C\mathbf{P}_j = {}_B T_G(^G\mathbf{P}_j - {}^G\mathbf{r}) = [^C x_j \quad {}^C y_j \quad {}^C z_j]^T = \frac{^C z_j}{f}[\mu_j \quad \nu_j \quad f]^T. \tag{5}$$

When considering the flight environment for fixed-wing aircraft, which generally holds the flight height above hundreds of meters, flat terrain such as plain and farm field the topographic relief can be neglected compared to the flight height. In such scenes, we assume that the visual features lie in a level plane. Further, without loss of generality, by assuming the feature plane contains the origin of the global inertial frame, we obtained $\mathbf{e}_3^{T\,G}\mathbf{P}_j = 0$, where $\mathbf{e}_3 = \begin{bmatrix} 0 & 0 & 1 \end{bmatrix}^T$. We also adopted the unit quaternion $_B\mathbf{q}_G = q_0\mathbf{i} + q_1\mathbf{j} + q_2\mathbf{k} + q_3$ to represent the rotation from the global coordinate frame $\{G\}$ to the body-fixed coordinate frame $\{B\}$ at the time t. Hence, $_B T_G$ can be replaced

by $C(_B\mathbf{q}_G)$ which describes the rotational matrix corresponding to $_B\mathbf{q}_G$. Thus, the scale of line-of-sight vector and also the depth component of $^C\mathbf{P}_j$, namely ^Cz_j in Eq. (5) can be represented as

$$^Cz_j = \frac{-\mathbf{e}_3^{T\,G}\mathbf{r}}{\mathbf{e}_3^T C^T(_B\mathbf{q}_G)\frac{1}{f}[\mu_j \quad \nu_j \quad f]^T}. \tag{6}$$

2.2 Optical Flow Computation

After the characterization of the motion equation of the camera and pinhole camera model, we focused on the optical flow computation method. In this work, the optical flow was determined as the displacement between two successive images divided by the time interval between their capture. Based on the local brightness constancy assumption and small motion assumption, the block-matching algorithm approximates the image motion by a displacement, which yielded the best match between image regions. The concept of the procedure, shown in Fig. 2, as well as the specific algorithm for displacement computation outlined in Fig. 3, starts with distinct feature extraction and ends up with minimizing the SSD cost function among the search window,[9,17] and the result provided the displacement $\mathbf{d}(\mathbf{X}_k)$ of the current feature, which can be formulated as

$$\text{SSD}(\mathbf{X}_k, dx, dy)$$
$$= \sum_{i=x_k}^{x_k+b-1} \sum_{j=y_k}^{y_k+b-1} [I_O(i,j,t) - I_C(i+dx, j+dy, t+\Delta t)]^2, \tag{7}$$

$$\mathbf{d}(\mathbf{X}_k) = \operatorname*{arg\,min}_{-w \le dx, dy \le w} \text{SSD}(\mathbf{X}_k, dx, dy), \tag{8}$$

where $I_O(\cdot)$ and $I_C(\cdot)$ are both image intensity functions. Thus, the two-dimensional (2-D) optical flow $\dot{\zeta}$ can be obtained, i.e., $\dot{\zeta} = \frac{\mathbf{d}(\mathbf{X}_k)}{\Delta t}$.

Furthermore, we proved that the error of optical flow computation based on the SSD block-matching algorithm
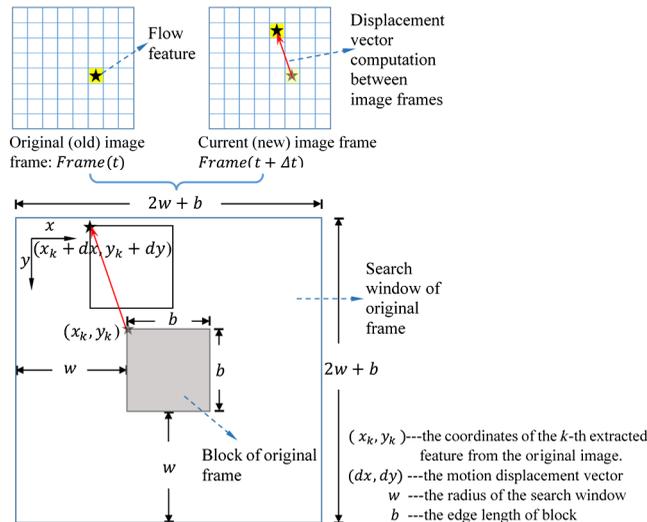

Fig. 2 Block-matching method.

Algorithm 1 for displacement computation

for all coordinates from the original image **do**
    set sum-of-squared differences (SSD) value and displacement(dx, dy) to zero
    if the feature is salient enough according to its gradient value of pixel intensity **then**
        obtain the distinct feature's coordinates $X_k = (x_k, y_k)$
        for each possible displacements: $-w \le dx, dy \le w$
            for $-x_k \le i \le x_k + b - 1$ and $-y_k \le j \le y_k + b - 1$
                calculate the SSD cost function expressed in Equation 7
                if the SSD value is smaller then before
                    update the SSD value and set $d(X_k) = (dx, dy)$
                end if
            end for
        end for
    end if
end for

Fig. 3 Algorithm for displacement computation.

is subject to a Gaussian distribution, which is not fully explained by most reviewed articles, such as Farid Kendoul's work.[9] Therefore, we redefined the whole algorithm as a maximum likelihood estimation problem. The displacement of the pixel can be seen as a warp function ω, which transforms a pixel point $\mathbf{p}_k(x_k, y_k)$ in the original image frame (I_O) into a point $\mathbf{p}_k'(x_k', y_k')$ in the current image frame (I_C) by a 2-D deformation vector ζ, namely $\mathbf{p}_k(x_k, y_k) \in I_O$, $\omega(\mathbf{p}_k, \zeta) = \mathbf{p}_k'(x_k', y_k') \in I_C$. Based on the local brightness constancy assumption, which is given as

$$I_O(\mathbf{p}_k) = I_C[\omega(\mathbf{p}_k, \zeta)] + \varepsilon_k, \tag{9}$$

without loss of generality, ε_k can be modeled as Gaussian noise as $\varepsilon_k \sim N(0, \sigma^2)$. Given the $b \times b$ size block of pixels, the problem aimed at the estimation of ζ by a patch of pixel samples $\{I_O(\mathbf{p}_k)\}_{k=1}^{n=b\times b}$. According to Eq. (9), $I_O(\mathbf{p}_k) \sim N\{I_C[\omega(\mathbf{p}_k, \zeta)], \sigma^2\}$, thus the corresponding probability density function is provided as

$$P[I_O(\mathbf{p}_k)|\zeta] = \frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{\{I_O(\mathbf{p}_k) - I_C[\omega(\mathbf{p}_k, \zeta)]\}^2}{2\sigma^2}\right). \tag{10}$$

Hence, we derived the likelihood function for the pixel samples as

$$L(\xi) = P[I_O(\mathbf{p}_1), I_O(\mathbf{p}_2) \cdots I_O(\mathbf{p}_N)|\zeta] = \prod_{k=1}^{n} P[I_O(\mathbf{p}_k)|\zeta]$$
$$= \prod_{k=1}^{n} C \cdot \exp\left(-\frac{\{I_O(\mathbf{p}_k) - I_C[\omega(\mathbf{p}_k, \zeta)]\}^2}{2\sigma^2}\right), \tag{11}$$

where the log-likelihood can be written as

$$\log L(\zeta) = -\frac{1}{2\sigma^2} \sum_{k=1}^{n} \{I_O(\mathbf{p}_k) - I_C[\omega(\mathbf{p}_k, \zeta)]\}^2 + n \log C, \tag{12}$$

in which, $n \log C$ only contains items irrelevant to ζ. The maximum likelihood estimator $\hat{\zeta}$ maximizes the log-likelihood, i.e., $\hat{\zeta} = \arg\max_\zeta \log L(\zeta)$, which is equivalent to finding the optimal displacement value that minimizes the least square function according to Eqs. (7) and (8). Therefore, we obtained $\hat{\zeta} = \operatorname*{arg\,min}_{-w \le dx, dy \le w} \text{SSD}(\mathbf{X}_k, dx, dy)$.

The maximum likelihood estimate (MLE) is proven to be asymptotically normally distributed,[18,19] and the estimated covariance error of MLE is given by the inverse of the observed Fisher information matrix; therefore, the maximum likelihood estimator can be expressed as

$$\hat{\zeta}_{\text{ML}} \overset{a}{\sim} N\left[\zeta_0, \frac{1}{n \cdot J(\zeta_0)}\right] = N\left[\zeta_0, \frac{1}{J(\hat{\zeta}_{\text{ML}})}\right], \tag{13}$$

where ζ_0 denotes the true displacement parameter, which can be obtained by the SSD block-matching algorithm. It is worth mentioning that in this context, the SSD block-matching algorithm is a brute force method that would guarantee returns with the global optimum. Matrix $J(\zeta_0)$ is the Fisher information for an individual observation, while $J(\hat{\zeta}_{\text{ML}})$ is the observed Fisher information matrix and is defined as

$$J(\hat{\zeta}_{\text{ML}}) = -E\left\{\left[\frac{\partial^2}{\partial x \partial y} \log L(\zeta)\right]\right\}, \tag{14}$$

which is also the expectation of the negative Hessian matrix of the log-likelihood function.[18] Since the optical flow $\dot{\zeta}$ was calculated with the displacement motion divided by inter-frame time, we saw that the optical flow error maintained a Gaussian distribution, which was the prerequisite for adopting the Kalman filter.

Next, we deduce the measurement equation of optical flow. The projected pixel coordinates of the feature point P_j on the image plane are given by Eq. (4), taking the time-derivatives of both sides of Eq. (4) leads to the expression for optical flow velocity

$$\dot{\zeta}_j = \begin{bmatrix} \dot{\mu}_j \\ \dot{\nu}_j \end{bmatrix} = \frac{f}{{}^C z_j^2} \begin{bmatrix} {}^C z_j & 0 & -{}^C x_j \\ 0 & {}^C z_j & -{}^C y_j \end{bmatrix} {}^C \dot{\mathbf{P}}_j. \tag{15}$$

Consider the relative motion between the camera and feature point by replacement of ${}^C \dot{\mathbf{P}}_j$ in Eq. (3) into Eq. (15), and eliminating $[{}^C x_j, {}^C y_j, {}^C z_j]$ with $[\mu_j, \nu_j, f]$ based on Eqs. (4) and (6), the optical flow equation can be computed as

$$\dot{\zeta}_j = \begin{bmatrix} \dot{\mu}_j \\ \dot{\nu}_j \end{bmatrix} = -\frac{\mathbf{e}_3^T C^T({}_B\mathbf{q}_G) \frac{1}{f} \begin{bmatrix} \mu_j \\ \nu_j \\ f \end{bmatrix}}{\mathbf{e}_3^T {}^G\mathbf{r}} \begin{bmatrix} -f & 0 & \mu_j \\ 0 & -f & \nu_j \end{bmatrix} C({}_B\mathbf{q}_G) {}^G\mathbf{v}$$
$$+ \begin{bmatrix} \frac{\mu_j \nu_j}{f} & -\left(f + \frac{\mu_j^2}{f}\right) & \nu_j \\ f + \frac{\nu_j^2}{f} & -\frac{\mu_j \nu_j}{f} & -\mu_j \end{bmatrix} {}^B\boldsymbol{\omega} + \boldsymbol{\xi}_j, \tag{16}$$

where ${}^G\mathbf{v}$ is the absolute velocity of the aircraft in the global inertial frame, i.e., ${}^G\dot{\mathbf{r}} = {}^G\mathbf{v}$, and $\boldsymbol{\xi}_j$ was modeled as zero mean, white Gaussian noise with covariance $\mathbf{R}_{\boldsymbol{\xi}_j}$ which is related to the inverse of the negative Hessian for log-likelihood function as shown in Eq. (14).

2.3 IMU Measurement

The IMU outputs are measured quantities in the body-fixed frame, angular rate $\boldsymbol{\omega}_m$ and acceleration $\mathbf{a}_m$, which were modeled as[20,21]

$$\boldsymbol{\omega}(t) = \boldsymbol{\omega}_m(t) - \mathbf{b}_{\boldsymbol{\omega}}(t) - \mathbf{n}_{\boldsymbol{\omega}}(t),$$
$$\dot{\mathbf{b}}_{\boldsymbol{\omega}}(t) = \mathbf{n}_{\mathbf{b}\boldsymbol{\omega}}(t),$$
$$\mathbf{a}(t) = C^T[{}_B\mathbf{q}_G(t)][\mathbf{a}_m(t) - \mathbf{b}_{\mathbf{a}}(t) - \mathbf{n}_{\mathbf{a}}(t)] + \mathbf{g},$$
$$\dot{\mathbf{b}}_{\mathbf{a}}(t) = \mathbf{n}_{\mathbf{ba}}(t), \tag{17}$$

where $\boldsymbol{\omega} \in \mathbb{R}^3$ and $\mathbf{a} \in \mathbb{R}^3$ are the true value, which are corrupted by IMU biases $\mathbf{b}_{\boldsymbol{\omega}}$ and $\mathbf{b}_{\mathbf{a}}$, IMU drift noise $\mathbf{n}_{\boldsymbol{\omega}}$ and $\mathbf{n}_{\mathbf{a}}$ which assumed to be zero-mean Gaussian white noise. $\mathbf{g}$ is the gravitational acceleration. The IMU biases are nonstatic but rather are seen as a random walk process with the biases noise $\mathbf{n}_{\mathbf{b}\boldsymbol{\omega}}$ and $n_{\mathbf{ba}}$. All the noise terms $\mathbf{n} = [\mathbf{n}_{\mathbf{a}}^T, \mathbf{n}_{\boldsymbol{\omega}}^T, \mathbf{n}_{\mathbf{ba}}^T, \mathbf{n}_{\mathbf{b}\boldsymbol{\omega}}^T]^T$ are specified by the corresponding scalar variance coefficient $\sigma_{\mathbf{n}_{\mathbf{a}}}^2$, $\sigma_{\mathbf{n}_{\boldsymbol{\omega}}}^2$, $\sigma_{\mathbf{n}_{\mathbf{ba}}}^2$, and $\sigma_{\mathbf{n}_{\mathbf{b}\boldsymbol{\omega}}}^2$; the diagonal covariance matrices are given by $\mathbf{Q}_{\mathbf{a}} = \sigma_{\mathbf{n}_{\mathbf{a}}}^2 \mathbf{I}_3$, $\mathbf{Q}_{\boldsymbol{\omega}} = \sigma_{\mathbf{n}_{\boldsymbol{\omega}}}^2 \mathbf{I}_3$, $\mathbf{Q}_{\mathbf{ba}} = \sigma_{\mathbf{n}_{\mathbf{ba}}}^2 \mathbf{I}_3$, and $\mathbf{Q}_{\mathbf{b}\boldsymbol{\omega}} = \sigma_{\mathbf{n}_{\mathbf{b}\boldsymbol{\omega}}}^2 \mathbf{I}_3$. According to Refs. 22 and 23, they can be obtained by the Allan variance technique.

3 Estimator Description

With the purpose to integrate the two different sources of information collected by IMU and monocular-camera, an ESKF was designed. While the IMU information served to make predictions to the filter, the optical flow vision information was used to correct the filter. Compared to the general extend Kalman filter, which consists of states prediction step and update step, the ESKF considers the error states induced by the reset process, producing more precise results by reducing the long-term error drift in odometry system.

3.1 System State and Propagation

The estimator vector was the 16×1 vector, and all the symbol definition can be found in Sec. 2.

$$\mathbf{x} \triangleq [{}^G\mathbf{r}^T \quad {}^G\mathbf{v}^T \quad {}_B\mathbf{q}_G^T \quad \mathbf{b}_{\mathbf{a}}^T \quad \mathbf{b}_{\boldsymbol{\omega}}^T]^T. \tag{18}$$

The system kinematic equation that describes the time evolution of the state was similar to Hesch's model,[21] of which the corresponding linear discretized form of predicted state estimate is provided as

$$
{}^G\hat{\mathbf{r}}_{k+1}^- = {}^G\hat{\mathbf{r}}_k^+ + \Delta t {}^G\hat{\mathbf{v}}_k^+ + \frac{\Delta t^2}{2}[\hat{C}_k^{+T}(\mathbf{a}_{m,k} - \hat{\mathbf{b}}_{\mathbf{a},k}^+) + \mathbf{g}],
$$
$$
{}^G\hat{\mathbf{v}}_{k+1}^- = {}^G\hat{\mathbf{v}}_k^+ + \Delta t[\hat{C}_k^{+T}(\mathbf{a}_{m,k} - \hat{\mathbf{b}}_{\mathbf{a},k}^+) + \mathbf{g}],
$$
$$
{}_B\hat{\mathbf{q}}_{G,k+1}^- = {}_B\hat{\mathbf{q}}_{G,k}^+ + \frac{\Delta t}{2}\Omega(\boldsymbol{\omega}_{m,k} - \hat{\mathbf{b}}_{\boldsymbol{\omega},k}^+) \cdot {}_B\hat{\mathbf{q}}_{G,k}^+,
$$
$$
\hat{\mathbf{b}}_{\mathbf{a},k+1}^- = \hat{\mathbf{b}}_{\mathbf{a},k}^+,
$$
$$
\hat{\mathbf{b}}_{\boldsymbol{\omega},k+1}^- = \hat{\mathbf{b}}_{\boldsymbol{\omega},k}^+. \tag{19}
$$

In order to derive the process noise covariance matrix, the error dynamic needed to be derived from the kinematic equation along its nominal form. We defined the error state in the estimate $\hat{x}$ of a quantity x for aircraft position, velocity, and IMU biases as $\delta x = x - \hat{x}$. And it is exceptive for the quaternion error, which was defined by the 3×1 angle-error vector $\delta\boldsymbol{\theta}$, as $\delta\mathbf{q} \triangleq [\frac{1}{2}\delta\boldsymbol{\theta}^T \quad 1]^T$. Then, the linearized first-order approximation of continuous-time error-state kinematics[14,20,24] are provided as

$$\delta\dot{\mathbf{r}}(t) = \delta\mathbf{v}(t),$$

$$\delta\dot{\mathbf{v}}(t) \approx -C^T({}_B\hat{\mathbf{q}}_G(t))\lfloor[\mathbf{a}_m(t) - \hat{\mathbf{b}}_\mathbf{a}(t)]\times\rfloor\delta\boldsymbol{\theta}(t)$$
$$- C^T[{}_B\hat{\mathbf{q}}_G(t)]\delta\mathbf{b}_\mathbf{a}(t) - C^T[{}_B\hat{\mathbf{q}}_G(t)]\mathbf{n}_\mathbf{a}(t),$$

$$\delta\dot{\boldsymbol{\theta}}(t) \approx -\lfloor[\boldsymbol{\omega}_m(t) - \hat{\mathbf{b}}_\boldsymbol{\omega}(t)]\times\rfloor\delta\boldsymbol{\theta}(t) - \delta\mathbf{b}_\boldsymbol{\omega} - \mathbf{n}_\boldsymbol{\omega}(t),$$

$$\delta\dot{\mathbf{b}}_\mathbf{a}(t) = \mathbf{n}_\mathbf{ba}(t),$$

$$\delta\dot{\mathbf{b}}_\boldsymbol{\omega}(t) = \mathbf{n}_\mathbf{b\omega}(t). \tag{20}$$

In practical terms, the ESKF is realized in a time-discrete manner, with the update time interval $\Delta t = t_{k+1} - t_k$, which was also seen as the sampling period of IMU signals. By using numerical integration of Eq. (20), we obtained the discrete error-state equation

$$\delta\mathbf{x} = [\,\delta\mathbf{r}^T \quad \delta\mathbf{v}^T \quad \delta\boldsymbol{\theta}^T \quad \delta\mathbf{b}_\mathbf{a}^T \quad \delta\mathbf{b}_\boldsymbol{\omega}^T\,]^T,$$
$$\delta\mathbf{x}_{k+1} = \boldsymbol{\Phi}_k\delta\mathbf{x}_k + \mathbf{G}_k\mathbf{n}_k, \tag{21}$$

where $\boldsymbol{\Phi}_k$ was the transition matrix and $\mathbf{G}_k$ denoted the perturbation matrix of noise vector $\mathbf{n}_k$. In the process of integration and discretization, based on Joan Sola's results,[20] the following auxiliary series was introduced:

$$\boldsymbol{\Gamma}_n^T(\boldsymbol{\omega}) = \sum_{k=0}^{\infty} \frac{\Delta t^{k+n}}{(k+n)!}\lfloor-\boldsymbol{\omega}\times\rfloor^k, \tag{22}$$

by assigning the value of n with $\{0,1,2,3\}$, we obtained

$$\boldsymbol{\Gamma}_0^T(\boldsymbol{\omega}) = \sum_{k=0}^{\infty} \frac{(\lfloor-\boldsymbol{\omega}\times\rfloor\Delta t)^k}{k!} = e^{\lfloor-\boldsymbol{\omega}\times\rfloor\Delta t} = R\{\boldsymbol{\omega}\Delta t\}^T,$$

$$\boldsymbol{\Gamma}_1^T(\boldsymbol{\omega}) = I\cdot\Delta t - \frac{(\lfloor-\boldsymbol{\omega}\times\rfloor)}{\|\boldsymbol{\omega}\|^2}\cdot[R\{\boldsymbol{\omega}\Delta t\}^T - I - (\lfloor-\boldsymbol{\omega}\times\rfloor\Delta t)],$$

$$\boldsymbol{\Gamma}_2^T(\boldsymbol{\omega}) = \frac{1}{2}I\cdot\Delta t^2 - \frac{1}{\|\boldsymbol{\omega}\|^2}\cdot\left[R\{\boldsymbol{\omega}\Delta t\}^T - I - (\lfloor-\boldsymbol{\omega}\times\rfloor\Delta t)\right.$$
$$\left. - \frac{1}{2}(\lfloor-\boldsymbol{\omega}\times\rfloor\Delta t)^2\right],$$

$$\boldsymbol{\Gamma}_3^T(\boldsymbol{\omega}) = \frac{1}{3!}I\cdot\Delta t^3 + \frac{(\lfloor-\boldsymbol{\omega}\times\rfloor)}{\|\boldsymbol{\omega}\|^2}\cdot\left[R\{\boldsymbol{\omega}\Delta t\}^T - I - (\lfloor-\boldsymbol{\omega}\times\rfloor\Delta t)\right.$$
$$\left. - \frac{1}{2}(\lfloor-\boldsymbol{\omega}\times\rfloor\Delta t)^2 - \frac{1}{3!}(\lfloor-\boldsymbol{\omega}\times\rfloor\Delta t)^3\right]. \tag{23}$$

Here, $\boldsymbol{\Gamma}_0^T$ means the incremental rotation matrix if rotating an arbitrary coordinate frame with a rotational rate of $-\boldsymbol{\omega}$ for timespan Δt according to the Rodrigues rotation formula; therefore, the discrete linearized error dynamic transition matrix $\boldsymbol{\Phi}_k$ as a function of above series[24] (for readability $\hat{\boldsymbol{\Gamma}}_n^T = \boldsymbol{\Gamma}_n^T[\boldsymbol{\omega}_m(k) - \hat{\mathbf{b}}_\boldsymbol{\omega}(k)]$) can be written as in matrix

$$\boldsymbol{\Phi}_k = \begin{bmatrix} \mathbf{I}_3 & \Delta t\mathbf{I}_3 & -\frac{\Delta t^2}{2}C^T[{}_B\hat{\mathbf{q}}_G(k)]\lfloor[\mathbf{a}_m(k) - \hat{\mathbf{b}}_\mathbf{a}(k)]\times\rfloor & -\frac{\Delta t^2}{2}C^T[{}_B\hat{\mathbf{q}}_G(k)] & \mathbf{0}_{3\times3} \\ \mathbf{0}_{3\times3} & \mathbf{I}_3 & -\Delta t C^T[{}_B\hat{\mathbf{q}}_G(k)]\lfloor[\mathbf{a}_m(k) - \hat{\mathbf{b}}_\mathbf{a}(k)]\times\rfloor & -\Delta t C^T[{}_B\hat{\mathbf{q}}_G(k)] & \mathbf{0}_{3\times3} \\ \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \hat{\boldsymbol{\Gamma}}_0^T & \mathbf{0}_{3\times3} & -\hat{\boldsymbol{\Gamma}}_1^T \\ \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{I}_3 & \mathbf{0}_{3\times3} \\ \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{I}_3 \end{bmatrix}, \tag{24}$$

$$\mathbf{G}_k = \begin{bmatrix} \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} \\ -\mathbf{I}_3 & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} \\ \mathbf{0}_{3\times3} & -\mathbf{I}_3 & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} \\ \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{I}_3 & \mathbf{0}_{3\times3} \\ \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{I}_3 \end{bmatrix}. \tag{25}$$

The discrete process noise covariance matrix $\mathbf{Q}_k$ was calculated as

$$\mathbf{Q}_k = \begin{bmatrix} \frac{\Delta t^3}{3}\mathbf{Q}_\mathbf{a} + \frac{\Delta t^5}{20}\mathbf{Q}_\mathbf{ba} & \frac{\Delta t^2}{2}\mathbf{Q}_\mathbf{a} + \frac{\Delta t^4}{8}\mathbf{Q}_\mathbf{ba} & \mathbf{0}_{3\times3} & -\frac{\Delta t^3}{6}C^T[{}_B\hat{\mathbf{q}}_G(k)]\mathbf{Q}_\mathbf{ba} & \mathbf{0}_{3\times3} \\ \frac{\Delta t^2}{2}\mathbf{Q}_\mathbf{a} + \frac{\Delta t^4}{8}\mathbf{Q}_\mathbf{ba} & \Delta t\mathbf{Q}_\mathbf{a} + \frac{\Delta t^3}{3}\mathbf{Q}_\mathbf{ba} & \mathbf{0}_{3\times3} & -\frac{\Delta t^2}{2}C^T[{}_B\hat{\mathbf{q}}_G(k)]\mathbf{Q}_\mathbf{ba} & \mathbf{0}_{3\times3} \\ \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \Delta t\mathbf{Q}_\boldsymbol{\omega} + (\hat{\boldsymbol{\Gamma}}_3^T + \hat{\boldsymbol{\Gamma}}_3)\cdot\mathbf{Q}_\mathbf{b\omega} & \mathbf{0}_{3\times3} & -\hat{\boldsymbol{\Gamma}}_2^T\cdot\mathbf{Q}_\mathbf{b\omega} \\ -\frac{\Delta t^3}{6}C[{}_B\hat{\mathbf{q}}_G(k)]\mathbf{Q}_\mathbf{ba} & -\frac{\Delta t^2}{2}C[{}_B\hat{\mathbf{q}}_G(k)]\mathbf{Q}_\mathbf{ba} & \mathbf{0}_{3\times3} & \Delta t\mathbf{Q}_\mathbf{ba} & \mathbf{0}_{3\times3} \\ \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & -\hat{\boldsymbol{\Gamma}}_2\cdot\mathbf{Q}_\mathbf{b\omega} & \mathbf{0}_{3\times3} & \Delta t Q_\mathbf{b\omega} \end{bmatrix}. \tag{26}$$

Thus, the error covariance propagation resulted as

$$\mathbf{P}_{k+1} = \boldsymbol{\Phi}_k\mathbf{P}_k\boldsymbol{\Phi}_k^T + \mathbf{Q}_k. \tag{27}$$

3.2 Measurement Update Equation

According to Eq. (16), the optical flow measurement can be expressed by the estimated state vector. Since the aircraft angular rate was not part of the state vector, we replaced

it with $^B\hat{\boldsymbol{\omega}} = \boldsymbol{\omega}_m(t) - \hat{\mathbf{b}}_{\boldsymbol{\omega}}(t)$. Substituting the coefficient matrices associating with μ_j, ν_j, and f in Eq. (16), and it would be clearer with notation listed as

$$\pi_{\mathbf{P_j}} \triangleq \frac{1}{f}[\mu_j \quad \nu_j \quad f]^T,$$

$$A_{\mathbf{P_j}} \triangleq \begin{bmatrix} -f & 0 & \mu_j \\ 0 & -f & \nu_j \end{bmatrix},$$

$$B_{\mathbf{P_j}} \triangleq \begin{bmatrix} \frac{\mu_j\nu_j}{f} & -\left(f + \frac{\mu_j^2}{f}\right) & \nu_j \\ f + \frac{\nu_j^2}{f} & -\frac{\mu_j\nu_j}{f} & -\mu_j \end{bmatrix}, \tag{28}$$

$$\hat{\dot{\boldsymbol{\zeta}}}_j = -\frac{\mathbf{e}_3^T C^T(_B\hat{\mathbf{q}}_G)\pi_{\mathbf{P_j}}}{\mathbf{e}_3^{T\,G}\hat{\mathbf{r}}}A_{\mathbf{P_j}}C(_B\hat{\mathbf{q}}_G)^G\hat{\mathbf{v}} + B_{\mathbf{P_j}}(\boldsymbol{\omega}_m - \hat{\mathbf{b}}_{\boldsymbol{\omega}}). \tag{29}$$

Furthermore, by subtracting measured angular rate item and redefining the revised optical flow measurement $\hat{\dot{\boldsymbol{\zeta}}}_j$ as a new measurement of the filter, the right side was expressed only using the state vector $\hat{\mathbf{x}}$; therefore, the following notation remains as

$$\hat{\dot{\tilde{\boldsymbol{\zeta}}}}_j = \hat{\dot{\boldsymbol{\zeta}}}_j - B_{\mathbf{P_j}}\boldsymbol{\omega}_m$$

$$= -\frac{\mathbf{e}_3^T C^T(_B\hat{\mathbf{q}}_G)\pi_{\mathbf{P_j}}}{\mathbf{e}_3^{T\,G}\hat{\mathbf{r}}}A_{\mathbf{P_j}}C(_B\hat{\mathbf{q}}_G)^G\hat{\mathbf{v}} - B_{\mathbf{P_j}}\hat{\mathbf{b}}_{\boldsymbol{\omega}}. \tag{30}$$

So that the measurement residual, also called innovation, was given as

$$\mathbf{y}_k^j = (\dot{\boldsymbol{\zeta}}_k^j - B_{\mathbf{P_j}}\boldsymbol{\omega}_m) - \hat{\dot{\tilde{\boldsymbol{\zeta}}}}_k^j \approx \mathbf{H}_k^j \delta\mathbf{x}_k + \boldsymbol{\eta}_k^j, \tag{31}$$

where the Jacobian measurement was

$$\mathbf{H}_k^j \triangleq \frac{\partial h^j}{\partial \mathbf{x}} \cdot \frac{\partial \mathbf{x}}{\partial \delta\mathbf{x}}\bigg|_{\mathbf{x}_k} = [\,\mathbf{H}_{\delta\mathbf{r},k}^j \quad \mathbf{H}_{\delta\mathbf{v},k}^j \quad \mathbf{H}_{\delta\boldsymbol{\theta},k}^j \quad 0_{2\times3} \quad \mathbf{H}_{\delta\mathbf{b}_{\boldsymbol{\omega}},k}^j\,], \tag{32}$$

$$\mathbf{H}_{\delta\mathbf{r},k}^j = \frac{\mathbf{e}_3^T C^T(_B\hat{\mathbf{q}}_{G,k})\pi_{\mathbf{P_j}}}{\mathbf{e}_3^{T\,G}\hat{\mathbf{r}}_k\,^G\hat{\mathbf{r}}_k^T\mathbf{e}_3}A_{\mathbf{P_j}}C(_B\hat{\mathbf{q}}_{G,k})^G\hat{\mathbf{v}}_k\mathbf{e}_3^T, \tag{33}$$

$$\mathbf{H}_{\delta\mathbf{v},k}^j = -\frac{\mathbf{e}_3^T C^T(_B\hat{\mathbf{q}}_{G,k})\pi_{\mathbf{P_j}}}{\mathbf{e}_3^{T\,G}\hat{\mathbf{r}}_k}A_{\mathbf{P_j}}C(_B\hat{\mathbf{q}}_{G,k}), \tag{34}$$

$$\mathbf{H}_{\delta\boldsymbol{\theta},k}^j = \frac{A_{\mathbf{P_j}}C(_B\hat{\mathbf{q}}_{G,k})^G\hat{\mathbf{v}}_k\mathbf{e}_3^T}{\mathbf{e}_3^{T\,G}\hat{\mathbf{r}}_k}C^T(_B\hat{\mathbf{q}}_{G,k})\lfloor\pi_{\mathbf{P_j}}\times\rfloor$$

$$-\frac{\mathbf{e}_3^T C^T(_B\hat{\mathbf{q}}_{G,k})\pi_{\mathbf{P_j}}}{\mathbf{e}_3^{T\,G}\hat{\mathbf{r}}_k}A_{\mathbf{P_j}}C(_B\hat{\mathbf{q}}_{G,k})\lfloor^G\hat{\mathbf{v}}_k\times\rfloor C^T(_B\hat{\mathbf{q}}_{G,k}), \tag{35}$$

$$\mathbf{H}_{\delta\mathbf{b}_{\boldsymbol{\omega}},k}^j = -B_{\mathbf{P_j}}. \tag{36}$$

Note that the residual noise $\boldsymbol{\eta}_k^j$ was involved with the optical flow algorithm error and gyroscope drift noise, i.e., $\boldsymbol{\eta}_k^j = \boldsymbol{\xi}_k^j - B_{\mathbf{P_j}} \cdot \mathbf{n}_{\boldsymbol{\omega},k}$, hence, the covariance matrix of

the revised optical flow measurement was the sum of the covariance of optical flow algorithm $\mathbf{R}_{\boldsymbol{\xi}_j}$ and the gyroscope covariance $\mathbf{Q}_{\boldsymbol{\omega}}$, which was given as

$$\mathbf{R}_k^j = E\{\boldsymbol{\eta}_k^j(\boldsymbol{\eta}_k^j)^T\} = \mathbf{R}_{\boldsymbol{\xi}_j} + B_{\mathbf{P_j}}\mathbf{Q}_{\boldsymbol{\omega}}B_{\mathbf{P_j}}^T. \tag{37}$$

In general, there were a set of optical flow vectors measured at one time, stacking all the observation residuals of Eq. (31) into one vector, which yielded, $\mathbf{y}_k = \mathbf{H}_k\delta\mathbf{x}_k + \boldsymbol{\eta}_k$, where $\mathbf{y}_k$, $\mathbf{H}_k$, and $\boldsymbol{\eta}_k$ were composed of the block elements $\mathbf{y}_k^j$, $\mathbf{H}_k^j$, and $\boldsymbol{\eta}_k^j$, such that the measurement covariance matrix could be stacked together diagonally as

$$\mathbf{R}_k = \begin{bmatrix} \mathbf{R}_k^1 & \cdots & 0_{3\times3} \\ \vdots & \ddots & \vdots \\ 0_{3\times3} & \cdots & \mathbf{R}_k^N \end{bmatrix}$$

$$= \begin{bmatrix} \mathbf{R}_{\boldsymbol{\xi}_1} + B_{\mathbf{P_1}}\mathbf{Q}_{\boldsymbol{\omega}}B_{\mathbf{P_1}}^T & \cdots & 0_{3\times3} \\ \vdots & \ddots & \vdots \\ 0_{3\times3} & \cdots & \mathbf{R}_{\boldsymbol{\xi}_N} + B_{\mathbf{P_N}}\mathbf{Q}_{\boldsymbol{\omega}}B_{\mathbf{P_N}}^T \end{bmatrix}. \tag{38}$$

Then, the *a priori* state estimate would be updated with current measurements according to the following steps: (1) calculate the Kalman gain as

$$\mathbf{K}_k = \mathbf{P}_k^-\mathbf{H}_k^T(\mathbf{H}_k\mathbf{P}_k^-\mathbf{H}_k^T + \mathbf{R}_k)^{-1}, \tag{39}$$

and (2) calculate the observed error state correction as

$$\Delta\mathbf{x}_k = \mathbf{K}_k\mathbf{y}_k, \tag{40}$$

with these quantities, the nominal states and error covariance get updated as

$$\hat{\mathbf{x}}_k^+ = \hat{\mathbf{x}}_k^- + \Delta\mathbf{x}_k,$$

$$\mathbf{P}_k^+ = (I - \mathbf{K}_k\mathbf{H}_k)\mathbf{P}_k^-(I - \mathbf{K}_k\mathbf{H}_k)^T + \mathbf{K}_k\mathbf{R}_k\mathbf{K}_k^T. \tag{41}$$

3.3 Error-State Kalman Filter Reset

For the ESKF reset process, we considered the expression among the new error $\delta\hat{\mathbf{x}}_k^+$, the old error $\delta\hat{\mathbf{x}}_k^-$, and the observed error correction $\Delta\mathbf{x}_k$, as the true value was constant, yielding $\hat{\mathbf{x}}_k^+ + \delta\hat{\mathbf{x}}_k^+ = \hat{\mathbf{x}}_k^- + \delta\hat{\mathbf{x}}_k^-$. Considering that the observed error has been injected into the updated nominal state, i.e., $\hat{\mathbf{x}}_k^+ = \hat{\mathbf{x}}_k^- + \Delta\mathbf{x}_k$ so that $\delta\hat{\mathbf{x}}_k^+ = \delta\hat{\mathbf{x}}_k^- - \Delta\mathbf{x}_k$, we defined an error reset function $\delta\mathbf{x} \triangleq g(\delta\mathbf{x}) = \delta\hat{\mathbf{x}}_k^- - \Delta\mathbf{x}_k$, where the reset operation was to ensure the error would reset $\delta\mathbf{x}\leftarrow0$, meanwhile, the covariance of error needed to be modified as well,[14] thus leading to

$$\mathbf{P}_k^+ \leftarrow G\mathbf{P}_k^+G^T, \tag{42}$$

where G represented the Jacobian matrix according to Joan Sola's proof[15] defined as

$$G \triangleq \frac{\partial g}{\partial \delta \mathbf{x}}\bigg|_{\delta \hat{\mathbf{x}}} = \begin{bmatrix} \mathbf{I}_6 & 0 & 0 \\ 0 & \mathbf{I}_3 - \lfloor \frac{1}{2} \delta \hat{\boldsymbol{\theta}} \times \rfloor & 0 \\ 0 & 0 & \mathbf{I}_6 \end{bmatrix}. \tag{43}$$

4 Observability Analysis

The estimator employed a linearized and discretized version of the nonlinear system model (to simplify the analysis, we only consider one single feature point):

$$\delta \mathbf{x}_{k+1} = \boldsymbol{\Phi}_k \delta \mathbf{x}_k + \mathbf{G}_k \mathbf{n}_k, \quad \mathbf{y}_k = \mathbf{H}_k \delta \mathbf{x}_k + \boldsymbol{\eta}_k, \tag{44}$$

where errors caused by linearization and discretization are combined with the noise items $\mathbf{n}_k$ and $\boldsymbol{\eta}_k$. With the purpose of revealing the theoretical limitation of the observer, a tool for the analysis of the observability properties—local observability matrix $\mathbf{M}(\mathbf{x})$ was adopted,[25] which was defined by the error dynamic transition matrix $\boldsymbol{\Phi}_k$ and measurement Jacobian $\mathbf{H}_k$, i.e.,

$$\mathbf{M}(\mathbf{x}) = \begin{bmatrix} \mathbf{H}_1 \\ \mathbf{H}_2 \boldsymbol{\Phi}_{2,1} \\ \vdots \\ \mathbf{H}_k \boldsymbol{\Phi}_{k,1} \end{bmatrix} = \begin{bmatrix} \mathbf{H}_1 \\ \mathbf{H}_2 \boldsymbol{\Phi}_1 \\ \vdots \\ \mathbf{H}_k \boldsymbol{\Phi}_{k-1} \boldsymbol{\Phi}_{k-2} \cdots \boldsymbol{\Phi}_1 \end{bmatrix}, \tag{45}$$

$$\mathbf{H}_k = \begin{bmatrix} \mathbf{H}_{\delta \mathbf{r},k} & \mathbf{H}_{\delta \mathbf{v},k} & \mathbf{H}_{\delta \boldsymbol{\theta},k} & 0_{2\times 3} & \mathbf{H}_{\delta \mathbf{b}_\omega,k} \end{bmatrix}, \tag{46}$$

$$\boldsymbol{\Phi}_{k,1} = \begin{bmatrix} \mathbf{I}_3 & \Delta t_k \mathbf{I}_3 & \boldsymbol{\Phi}^{13} & \boldsymbol{\Phi}^{14} & \boldsymbol{\Phi}^{15} \\ 0_{3\times 3} & \mathbf{I}_3 & \boldsymbol{\Phi}^{23} & \boldsymbol{\Phi}^{24} & \boldsymbol{\Phi}^{25} \\ 0_{3\times 3} & 0_{3\times 3} & \boldsymbol{\Phi}^{33} & 0_{3\times 3} & \boldsymbol{\Phi}^{35} \\ 0_{3\times 3} & 0_{3\times 3} & 0_{3\times 3} & \mathbf{I}_3 & 0_{3\times 3} \\ 0_{3\times 3} & 0_{3\times 3} & 0_{3\times 3} & 0_{3\times 3} & \mathbf{I}_3 \end{bmatrix}. \tag{47}$$

Here $\Delta t_k = t_k - t_1$, and the computing method for obtaining analogous expressions of the subblock elements $\boldsymbol{\Phi}^{mn}$ can be found in Joel Hesch's work.[26] Furthermore, using these expressions of $\boldsymbol{\Phi}^{mn}$, we obtained the k'th block row $\mathbf{M}_k$ of $\mathbf{M}(\mathbf{x})$, for $k \geq 2$

$$\mathbf{M}_k = \mathbf{H}_k \boldsymbol{\Phi}_{k,1} = \begin{bmatrix} \mathbf{M}_{k1} & \mathbf{M}_{k2} & \mathbf{M}_{k3} & \mathbf{M}_{k4} & \mathbf{M}_{k5} \end{bmatrix}, \tag{48}$$

and both $\boldsymbol{\Phi}^{mn}$ and $\mathbf{M}_k$ are presented in detail in Sec. 8. It is noted that $\mathbf{M}_{k4}$ and $\mathbf{M}_{k5}$, as containing the nonzero integrals are time-varying matrices and have the linearly independent columns. So that we can only consider the remaining block elements of $\mathbf{M}_k$ to form the basis of the right nullspace of $\mathbf{M}(\mathbf{x})$, which is stated as follows:

Theorem I: The right nullspace $\mathbf{N}$ of the local observability matrix $\mathbf{M}(\mathbf{x})$ for the linearized model is spanned by three unobservable directions:

$$\text{null}[\mathbf{M}(\mathbf{x})] = \mathbf{N} = \begin{bmatrix} \mathbf{N}_1 & 0_{3\times 1} \\ 0_{3\times 2} & -\lfloor {}^G \mathbf{v}_1 \times \rfloor \mathbf{g} \\ 0_{3\times 2} & \mathbf{C}_1 \mathbf{g} \\ 0_{3\times 2} & 0_{3\times 1} \\ 0_{3\times 2} & 0_{3\times 1} \end{bmatrix}, \quad \mathbf{N}_1 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 0 & 0 \end{bmatrix}. \tag{49}$$

The proof of the Theorem I is detailed in Sec. 9.

Nullspace $\mathbf{N}$ reveals that global rotations about the gravity vector is not observable, and global horizontal position is unobservable while height is fully observable. The remaining states, i.e., the velocity, pitch, and roll angle as well as IMU biases are also observable. The fact that the rotation around the gravity vector is unobservable is due to the missing of an absolute inertial yaw orientation, namely the yaw angle in the inertial frame is unobservable, while the absolute roll and pitch angle of aircraft w.r.t the global inertial frame is made observable by the gravity vector measurement. Similarly, without a global reference, the position in the direction of x and y is not observable, but what is different from previous findings, the altitude is observable because the assumption that optical flow features used for filter updates lie in a level plane and the plane contains the origin of the inertial frame. Only use the feature-plane assumption, according to Eq. (6), the feature depth is explicitly determined by absolute altitude and aircraft attitude, which becomes part of the components in the optical flow measurement equation and it, therefore, can get corrected when the new measurement update is coming.

5 Numerical Simulation

To numerically investigate the previous analysis, the proposed navigation scheme was implemented in a simulation. As shown in Fig. 4, there were 100 features randomly placed in a $[-350, 350 \text{ m}] \times [-350, 350 \text{ m}]$ horizontal plane around the origin. For generating the ground truth and sensor readings, we simulated a flight with a piecewise continuous trajectory, beginning with constant velocity at 20 m/s straight line motion lasting for 4 s, following by banking turn with a maximal angle of bank of 30 deg, and then a spiral motion with a pitch angle of up to 9 deg at the altitude between -200 and -300 m, ending with a 50 s of circling. A full list of parameters is shown in Table 1, in which most of those quantities were referred in the commonly used IMU datasheet, e.g., MPU6050 or ADIS16448.

The choices of the initial state values are given in Table 1, which is showing that the filter had fault tolerance to the initial state offset to an extent for convergence. In addition, the initial filter error covariance matrix was set as $\mathbf{P}_0 = \text{diag}([2500\mathbf{I}_3 \quad 100\mathbf{I}_3 \quad 0.25\mathbf{I}_3 \quad 0.01\mathbf{I}_3 \quad 7.6 \times 10^{-5}\mathbf{I}_3])$.
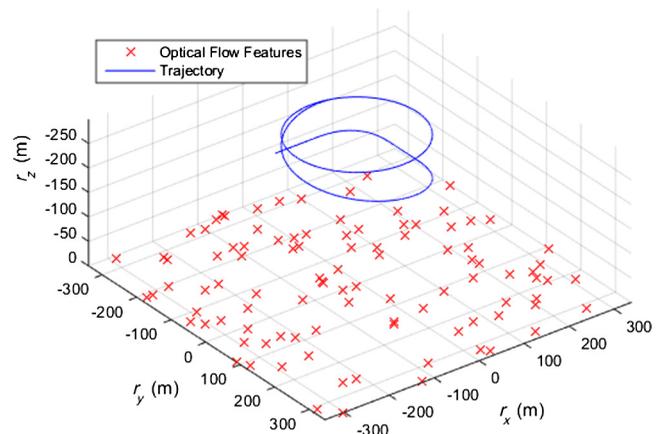

Fig. 4 IMU-camera trajectory and feature used in the simulation.

Table 1 Simulation parameters and initial conditions of the filter ["rand (3, 1)" here represents a 3×1 random vector obeying standard normal distribution].

Parameter	Value
IMU sampling rate (Hz)	100
Camera process rate (Hz)	30
Accelerometer noise [m/$\left(s\sqrt{s}\right)$]	$\sigma_{n_a} = 2.24 \times 10^{-3}$
Gyro noise [deg$/\sqrt{s}$]	$\sigma_{n_w} = 0.005$
Accelerometer bias noise [m/$\left(s^2\sqrt{s}\right)$]	$\sigma_{n_{ba}} = 7.53 \times 10^{-5}$
Gyro bias noise [rad/$(s\sqrt{s})$]	$\sigma_{n_{bw}} = 1.08 \times 10^{-5}$
Field of view (FOV) (deg)	90
Optical flow noise (pixel·rad/s)	0.01

State	Initial value	True value
Position (m)	True value + 50·rand(3,1)	[-50 -180 -200]
Velocity (m/s)	True value + 10·rand(3,1)	[20 0 0]
Quaternion	True value + 0.5·rand(3,1)	[0 0 0 1]
Accelerometer bias (m/s^2)	[0 0 0]	[0.0981 0.0981 0.0981]
Gyro bias (deg/s)	[0 0 0]	[0.5 0.5 -0.5]

Even though using a single optical flow measurement ensured the observability, in a practical trial through proper setting of FOV and initial feature quantity, we would have at least seven features to be extracted every time step to obtain better convergence. After running 100 Monte Carlo simulations, we computed the Monte Carlo standard deviation, also known as the root mean squared error, denoted by the sample error covariance P_j^{Monte} through collecting N simulation results as

$$P_j^{\text{Monte}} = \frac{1}{N}\sum_{i=1}^{N}(x_j - \hat{x}_{ij})(x_j - \hat{x}_{ij})^T. \qquad (50)$$

Here, at the j'th time step, x_j was the true state, and $\hat{x}_{ij}$ was the estimated value in the i'th simulation. The Monte Carlo standard deviation and filter standard deviation are the square root of the elements in the main diagonal of P_j^{Monte} and ESKF estimation error covariance, respectively.

In Figs. 5–9, it is shown the results of the proposed filter tracking ground truth for the position, attitude, velocity, and IMU bias. The shown results outline that most of the state estimations converge to ground truth occurred within 20 s, except the unobservable states, positions along x and y axes and the yaw angle, whose Monte Carlo standard deviation as well as the filter standard deviation at the final time maintained around values at the initial time. The unobservable states could remain with a constant error rather than absolute divergence since they could have proper evolution and IMU biases correction but lacking the initial reference. This correlated well with the observability analysis of horizontal position and yaw in Sec. 4.

Among the observable states, as the observability analysis disregards process noise and initial offset, as the filter standard deviation decreases promptly, the Monte Carlo standard
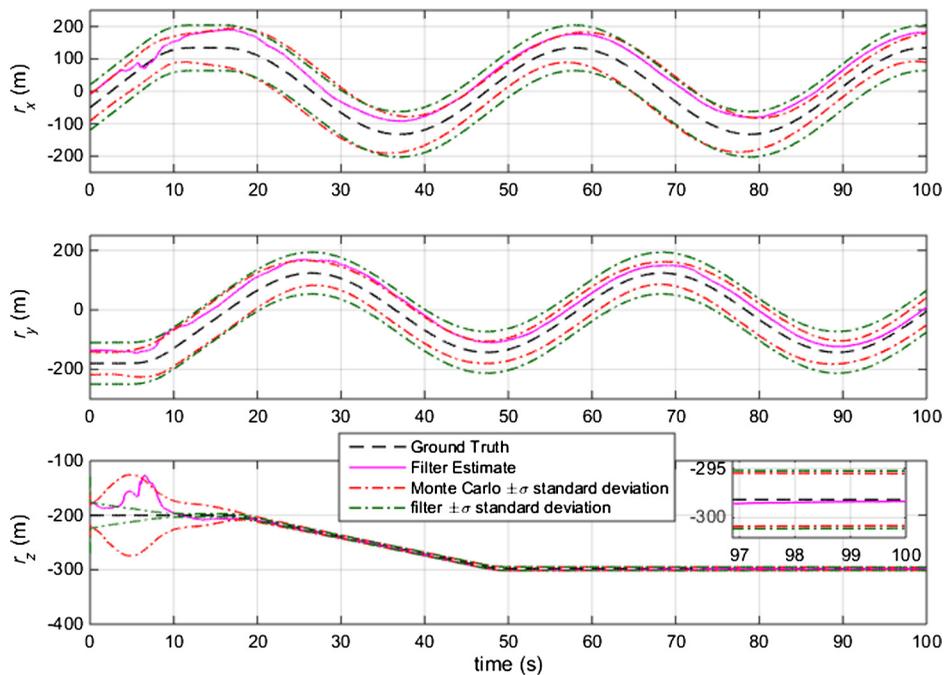

Fig. 5 Position estimation and corresponding Monte Carlo and filter standard deviation σ hull with respect to the ground truth.
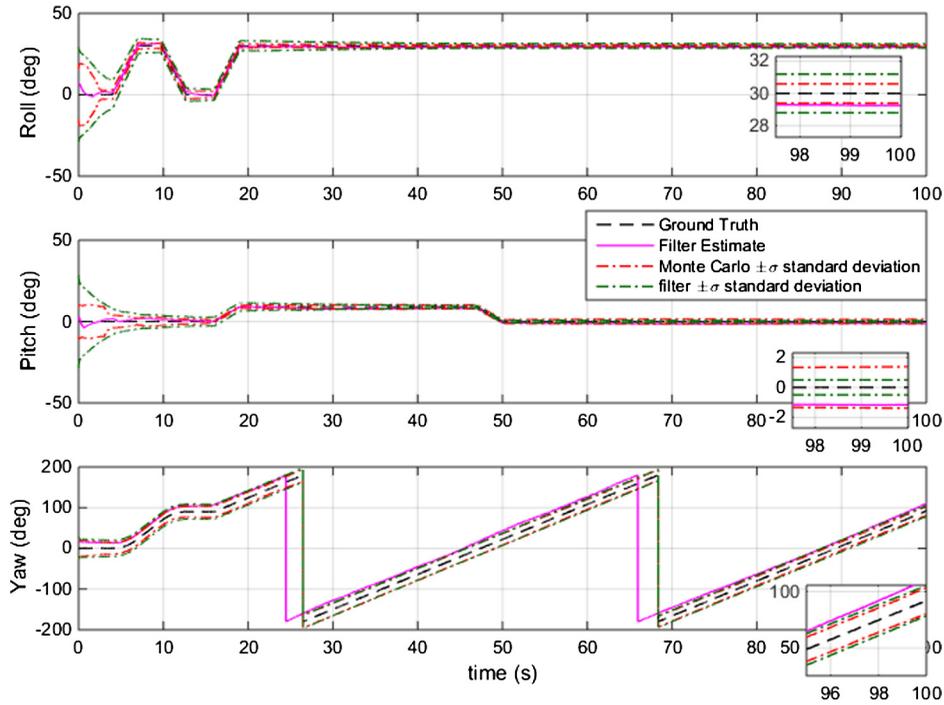
Fig. 6 Attitude estimation and corresponding Monte Carlo and filter standard deviation σ hull with respect to the ground truth.
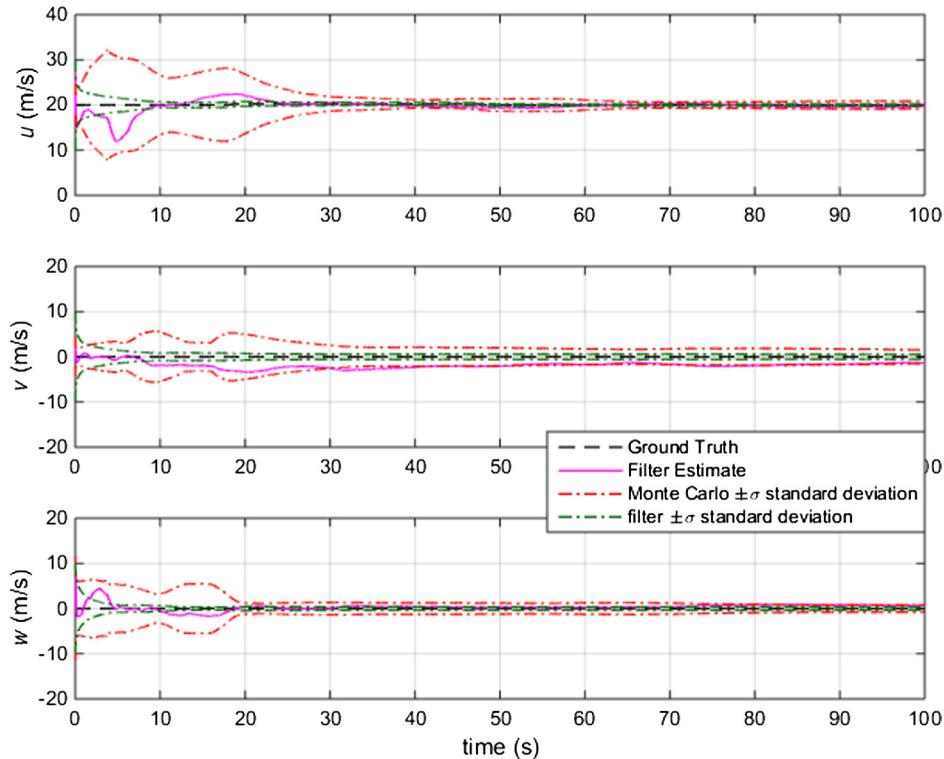

Fig. 7 Body-fixed velocity estimation and corresponding Monte Carlo and filter standard deviation σ hull with respect to the ground truth.

deviation seems to converge to zero slowly. Comparing with curves of the gyroscope bias, the uncertainty of accelerometer bias presented a much slower convergence, which was induced by the gyroscope bias being explicitly included in the measurement equation and therefore directly corrected while the accelerometer bias only relied on states propagation and obtained the effect via coupling with other states. For the estimation of roll and pitch angle, height as well
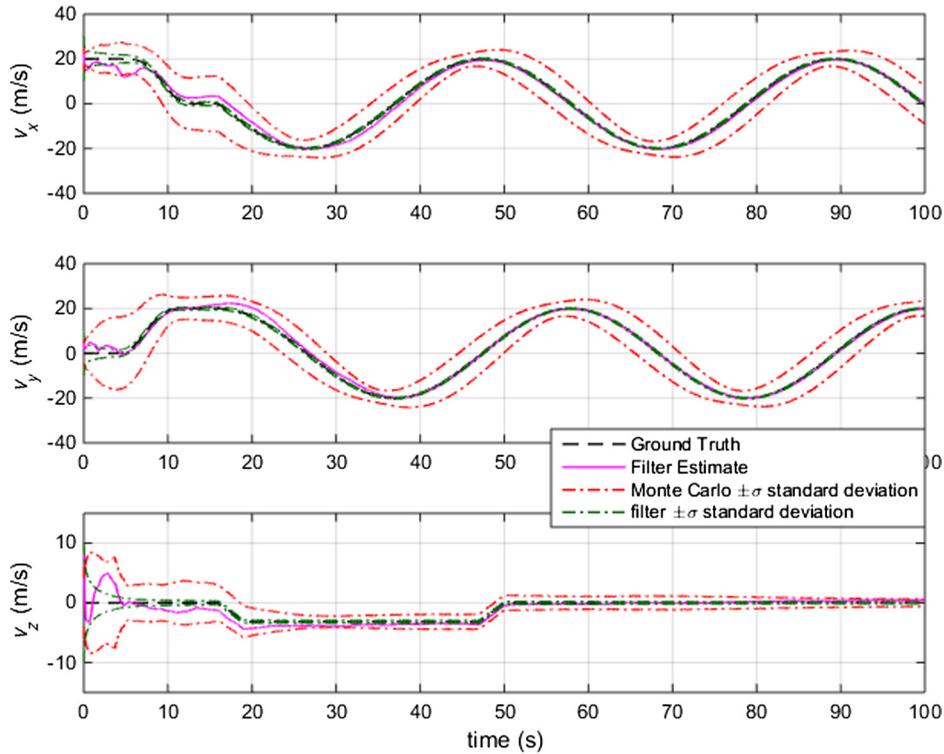
Fig. 8 Inertial velocity estimation and corresponding Monte Carlo and filter standard deviation σ hull with respect to the ground truth.
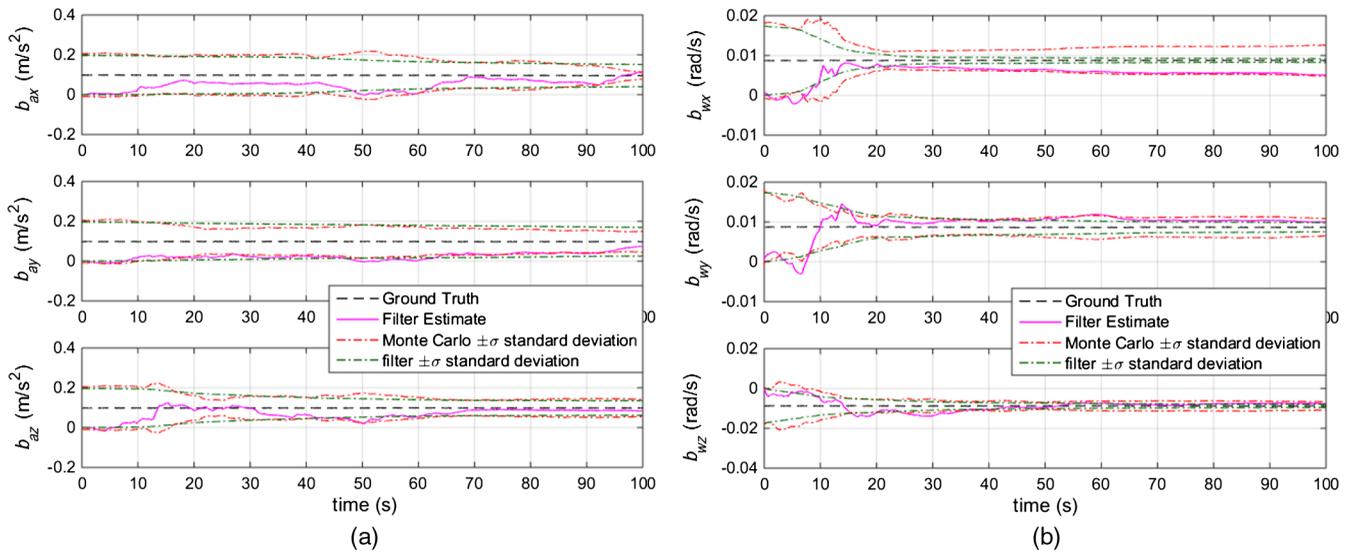

Fig. 9 IMU bias estimation and corresponding Monte Carlo and filter standard deviation σ hull with respect to the ground truth: (a) the accelerometer bias and (b) the gyro bias.

as vertical velocity, a better agreement between the Monte Carlo sample covariance and the filter error covariance than other states are shown in plots; nevertheless, the estimation of horizontal inertial and body-fixed velocity relied on mutual tight coupling effect with estimated attitude contained with accelerometer bias and unobservable yaw angle.

6 Experiment

The proposed filter has been tested on an X-star quadcopter which was equipped with Pixhawk 4 and PX4Flow board.

The optical flow camera consisted of a 752×480 MT9V034 image CMOS sensor with global shutter and 16 mm M12 lens, 24×24 μm, whose output to optical flow data was 4×4 binned image at 10 Hz. The camera module could be considered as a pinhole camera with the mounting method for pointing the camera straight down. The employed IMU was a dual redundancy solution with BMI055 and ICM-20689, providing inertial measurements at 125 Hz.

Ground truth was collected by the conventional INS based on extended Kalman filter and attitude and heading reference
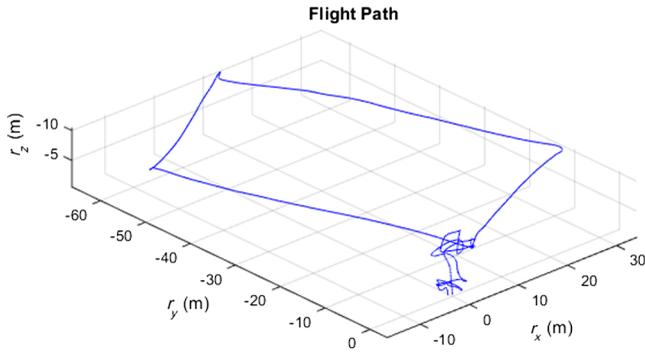
Flight Path

Fig. 10 Test flight path.

system algorithm, which integrated IMU outputs, magnetometer readings, and barometric altitude with GPS data. Note that these extra sensors were only for generating true value and not a part of the proposed navigation system, nor used for estimates.

In Fig. 10 the outdoor trajectory during the autonomous controlled waypoint mode flight test is shown. The flight envelope was set with a maximum height at 10 m and maximum speed around 7 m/s. The quadcopter took off at the origin, as we had little information of the IMU, the estimated states were initialized with zero value as this was the only option. The covariance parameters could be selected based on hardware specifications, which did not require strong tuning as the framework could work well with a large range of parameters.

From results in Figs. 11–13, the filter operation was started in the flight at around $t = 70$ s and at a speed of ~5 m/s. The roll and pitch angle exhibit a high-quality tracking accuracy except the growing error occurs when the vehicle was ready to descend for landing at the time after

150 s, as the horizontal velocity drops which was not favorable for the accuracy of optical flow measurement. Due to misalignment of the initial value, the yaw angle always kept a deviation with true value, which might make a major contribution to the estimated error of velocity along x and y axes, especially during $t \in [120, 140 \text{ s}]$ as they deviate from ground truth with a greater oscillation. Under unknown noise of the hardware system or vehicle vibration, the horizontal velocity and horizontal position estimates approached their true value more slowly and less precisely, but vertical velocity, as well as altitude estimate, exhibited more satisfactory accuracy.

7 Conclusions

This paper has presented an optical flow-based approach to vision-aided inertial navigation. Under the assumption that the environment of the observed feature points can be seen as located on a plane, the feature depth can be obtained by deducing distance expression through the optical flow equation with a geometry-coordinate relationship based on a vehicle-camera-feature system. In order to fill in previous works' gap, an error analysis for optical flow measurement using a block-matching method was provided, which was the prerequisite for adopting and designing an ESKF. With the IMU dead reckoning used for prediction, optical flow, and feature measurement for correction, the estimates of the vehicle's altitude, attitude, velocity, and drift IMU biases can be obtained. The theoretical limitations and feasibility of the filter were investigated by observability analysis, which has shown that only the absolute position along x and y axes along with the yaw angle were unobservable. Finally, the Monte Carlo simulation and hardware involved experiment validated the statement in observability analysis.

The numerical and real experiments have shown that the effectiveness and performance of the proposed filter may suffer from initialization errors and noise. Considering that
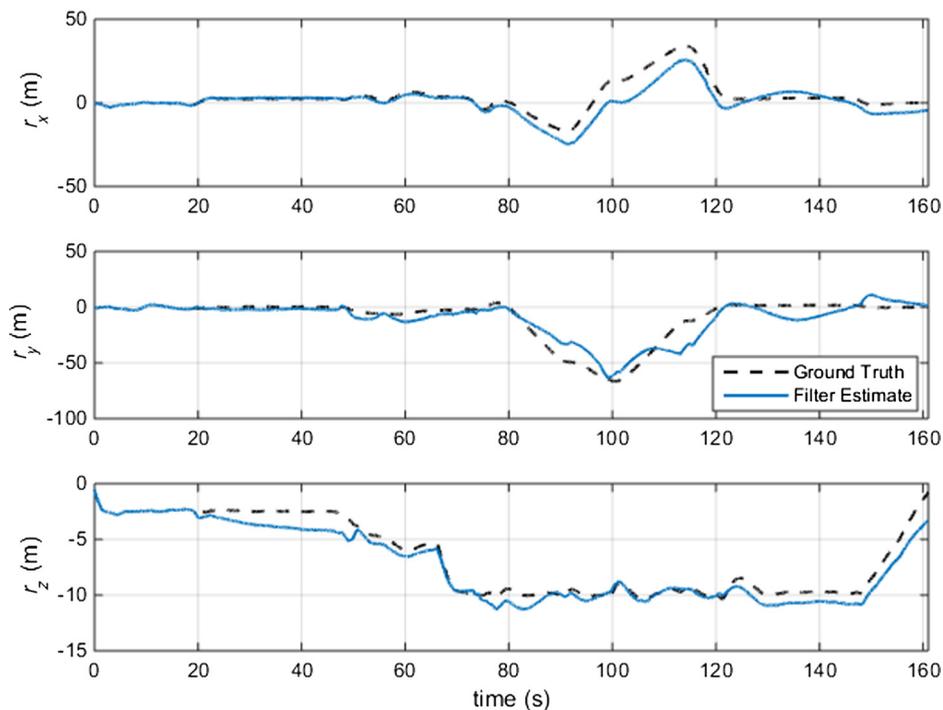

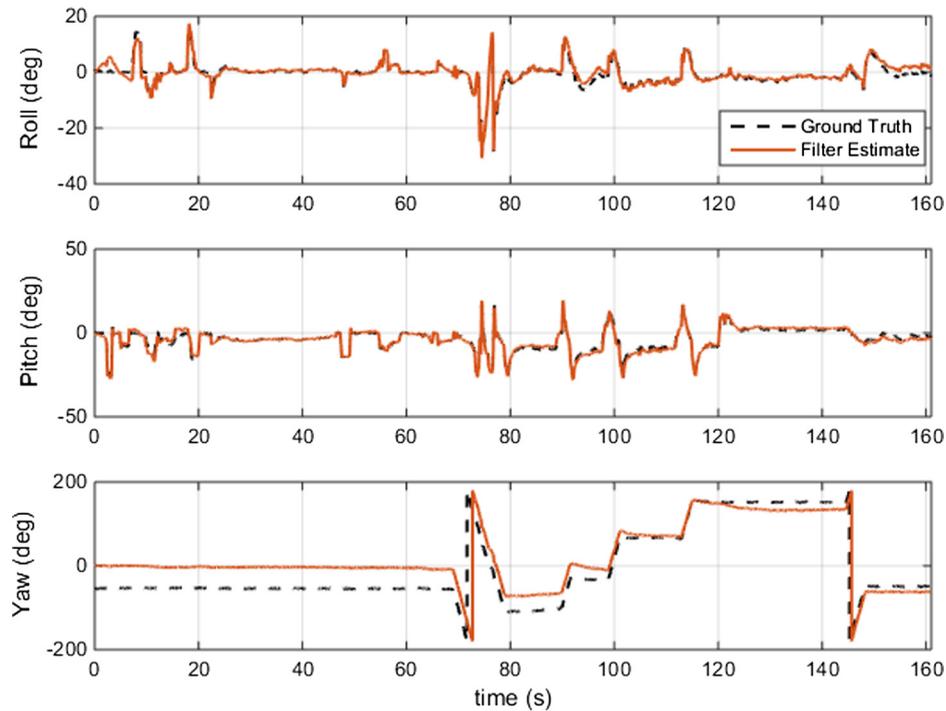
Fig. 11 Position in test flight.
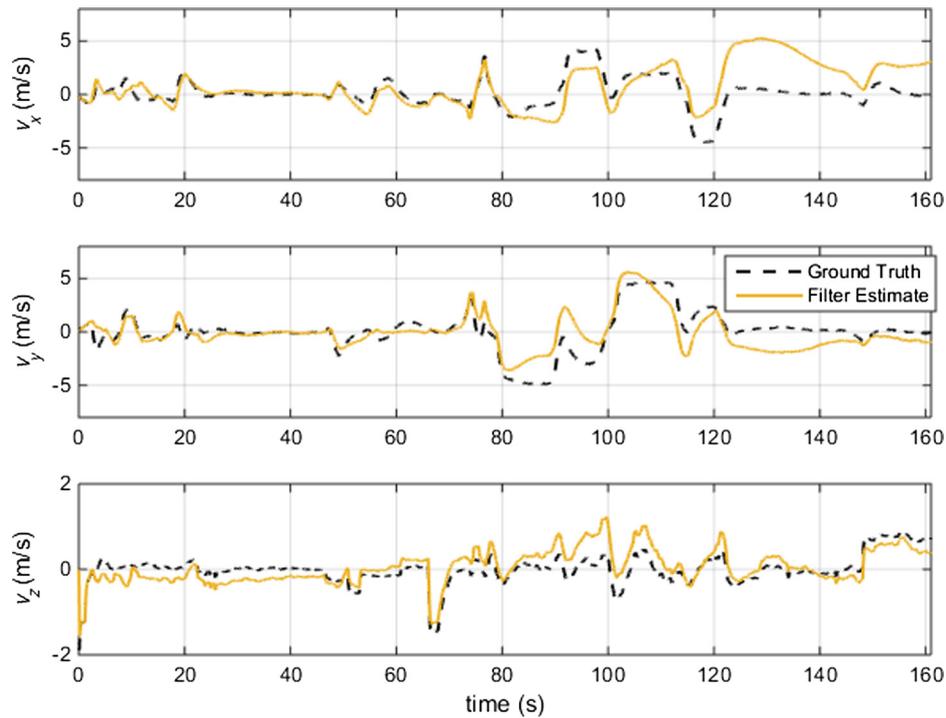
Fig. 12 Attitude in test flight.

Fig. 13 Inertial velocity in the test flight.

the minimal sensor configuration was low-priced and the adopted method required less computational effort, the overall system could be an onboard estimator for providing real-time supplemental state estimate when the UAV would be placed in an unknown GPS-denied environment.

8 Appendix I: Expression of Error Dynamic Transition Matrix Subblock Elements Φ^{mn} and Calculation of M_k

According to Ref. 26, the sub-block elements used in this paper are shown as

$$\boldsymbol{\Phi}^{13} = \left\lfloor \left({}^G\mathbf{r}_1 + {}^G\mathbf{v}_1\Delta t_k + \frac{1}{2}\mathbf{g}\Delta t_k^2 - {}^G\mathbf{r}_k \right)\times \right\rfloor \mathbf{C}_1^T$$

$$\boldsymbol{\Phi}^{23} = -\lfloor({}^G\mathbf{v}_k - {}^G\mathbf{v}_1 - \mathbf{g}\Delta t_k)\times\rfloor\mathbf{C}_1^T$$

$$\boldsymbol{\Phi}^{33} = \mathbf{C}_k\mathbf{C}_1^T$$

$$\boldsymbol{\Phi}^{14} = -\int_{t_1}^{t_k}\int_{t_1}^{t}\mathbf{C}_\tau^T\mathrm{d}\tau\,\mathrm{d}t$$

$$\boldsymbol{\Phi}^{24} = -\int_{t_1}^{t_k}\mathbf{C}_\tau^T\mathrm{d}\tau$$

$$\boldsymbol{\Phi}^{15} = \int_{t_1}^{t_k}\int_{t_1}^{t}\lfloor({}^G\mathbf{a}_s - \mathbf{g})\times\rfloor\int_{t_1}^{s}\mathbf{C}_\tau^T\mathrm{d}\tau\,\mathrm{d}s\,\mathrm{d}t$$

$$\boldsymbol{\Phi}^{25} = \int_{t_1}^{t_k}\lfloor({}^G\mathbf{a}_s - \mathbf{g})\times\rfloor\int_{t_1}^{s}\mathbf{C}_\tau^T\mathrm{d}\tau\,\mathrm{d}s$$

$$\boldsymbol{\Phi}^{35} = -\mathbf{C}_k\int_{t_1}^{t_k}\mathbf{C}_t^T\mathrm{d}t. \tag{51}$$

Moreover, with these expressions, and based on Eqs. (33)–(36), the k'th block row $\mathbf{M}_k$ of $\mathbf{M}(\mathbf{x})$ was calculated, for $k \geq 2$

$$\mathbf{M}_{k1} = \mathbf{H}_{\delta\mathbf{r},k} = \frac{\mathbf{e}_3^T\mathbf{C}_k^T\pi_{\mathbf{P}}}{\mathbf{e}_3^{T\,G}\mathbf{r}_k{}^G\mathbf{r}_k{}^T\mathbf{e}_3}A_{\mathbf{P}}\mathbf{C}_k{}^G\mathbf{v}_k\mathbf{e}_3^T, \tag{52}$$

$$\mathbf{M}_{k2} = \Delta t_k\mathbf{H}_{\delta\mathbf{r},k} + \mathbf{H}_{\delta\mathbf{v},k}$$
$$= \Delta t_k\frac{\mathbf{e}_3^T\mathbf{C}_k^T\pi_{\mathbf{P}}}{\mathbf{e}_3^{T\,G}\mathbf{r}_k{}^G\mathbf{r}_k{}^T\mathbf{e}_3}A_{\mathbf{P}}\mathbf{C}_k{}^G\mathbf{v}_k\mathbf{e}_3^T - \frac{\mathbf{e}_3^T\mathbf{C}_k^T\pi_{\mathbf{P}}}{\mathbf{e}_3^{T\,G}\mathbf{r}_k}A_{\mathbf{P}}\mathbf{C}_k, \tag{53}$$

$$\mathbf{M}_{k3} = \mathbf{H}_{\delta\mathbf{r},k}\boldsymbol{\Phi}^{13} + \mathbf{H}_{\delta\mathbf{v},k}\boldsymbol{\Phi}^{23} + \mathbf{H}_{\delta\boldsymbol{\theta},k}\boldsymbol{\Phi}^{33}$$
$$= \frac{\mathbf{e}_3^T\mathbf{C}_k^T\pi_{\mathbf{P}}}{\mathbf{e}_3^{T\,G}\mathbf{r}_k{}^G\mathbf{r}_k{}^T\mathbf{e}_3}A_{\mathbf{P}}\mathbf{C}_k{}^G\mathbf{v}_k\mathbf{e}_3^T$$
$$\cdot \left\lfloor \left({}^G\mathbf{r}_1 + {}^G\mathbf{v}_1\Delta t_k + \frac{1}{2}\mathbf{g}\Delta t_k^2 - {}^G\mathbf{r}_k \right)\times \right\rfloor\mathbf{C}_1^T$$
$$+ \frac{\mathbf{e}_3^T\mathbf{C}_k^T\pi_{\mathbf{P}}}{\mathbf{e}_3^{T\,G}\mathbf{r}_k}A_{\mathbf{P}}\mathbf{C}_k\cdot\lfloor({}^G\mathbf{v}_k - {}^G\mathbf{v}_1 - \mathbf{g}\Delta t_k)\times\rfloor\mathbf{C}_1^T$$
$$+ \frac{A_{\mathbf{P}}\mathbf{C}_k{}^G\mathbf{v}_k\mathbf{e}_3^T}{\mathbf{e}_3^{T\,G}\mathbf{r}_k}\lfloor\mathbf{C}_k^T\pi_{\mathbf{P}}\times\rfloor\mathbf{C}_1^T$$
$$- \frac{\mathbf{e}_3^T\mathbf{C}_k^T\pi_{\mathbf{P}}}{\mathbf{e}_3^{T\,G}\mathbf{r}_k}A_{\mathbf{P}}\mathbf{C}_k\lfloor{}^G\mathbf{v}_k\times\rfloor\mathbf{C}_1^T, \tag{54}$$

$$\mathbf{M}_{k4} = \mathbf{H}_{\delta\mathbf{r},k}\boldsymbol{\Phi}^{14} + \mathbf{H}_{\delta\mathbf{v},k}\boldsymbol{\Phi}^{24}$$
$$= \frac{\mathbf{e}_3^T\mathbf{C}_k^T\pi_{\mathbf{P}}}{\mathbf{e}_3^{T\,G}\mathbf{r}_k{}^G\mathbf{r}_k{}^T\mathbf{e}_3}A_{\mathbf{P}}\mathbf{C}_k{}^G\mathbf{v}_k\mathbf{e}_3^T\cdot\int_{t_1}^{t_k}\int_{t_1}^{t}\mathbf{C}_\tau^T\mathrm{d}\tau\,\mathrm{d}t$$
$$- \frac{\mathbf{e}_3^T\mathbf{C}_k^T\pi_{\mathbf{P}}}{\mathbf{e}_3^{T\,G}\mathbf{r}_k}A_{\mathbf{P}}\mathbf{C}_k\cdot\int_{t_1}^{t_k}\mathbf{C}_\tau^T\mathrm{d}\tau, \tag{55}$$

$$\mathbf{M}_{k5} = \mathbf{H}_{\delta\mathbf{r},k}\boldsymbol{\Phi}^{15} + \mathbf{H}_{\delta\mathbf{v},k}\boldsymbol{\Phi}^{25} + \mathbf{H}_{\delta\boldsymbol{\theta},k}\boldsymbol{\Phi}^{35} + \mathbf{H}_{\delta\mathbf{b}_{\boldsymbol{\omega}},k}$$
$$= -\frac{\mathbf{e}_3^T\mathbf{C}_k^T\pi_{\mathbf{P}}}{\mathbf{e}_3^{T\,G}\mathbf{r}_k{}^G\mathbf{r}_k{}^T\mathbf{e}_3}A_{\mathbf{P}}\mathbf{C}_k{}^G\mathbf{v}_k\mathbf{e}_3^T$$
$$\cdot \int_{t_1}^{t_k}\int_{t_1}^{t}\lfloor({}^G\mathbf{a}_s - \mathbf{g})\times\rfloor\int_{t_1}^{s}\mathbf{C}_\tau^T\mathrm{d}\tau\,\mathrm{d}s\,\mathrm{d}t$$
$$- \frac{\mathbf{e}_3^T\mathbf{C}_k^T\pi_{\mathbf{P}}}{\mathbf{e}_3^{T\,G}\mathbf{r}_k}A_{\mathbf{P}}\mathbf{C}_k\cdot\int_{t_1}^{t_k}\lfloor({}^G\mathbf{a}_s - \mathbf{g})\times\rfloor\int_{t_1}^{s}\mathbf{C}_\tau^T\mathrm{d}\tau\,\mathrm{d}s$$
$$- \frac{A_{\mathbf{P}}\mathbf{C}_k{}^G\mathbf{v}_k\mathbf{e}_3^T}{\mathbf{e}_3^{T\,G}\mathbf{r}_k}\mathbf{C}_k^T\lfloor\pi_{\mathbf{P}}\times\rfloor\mathbf{C}_k\int_{t_1}^{t_k}\mathbf{C}_t^T\mathrm{d}t$$
$$- \frac{\mathbf{e}_3^T\mathbf{C}_k^T\pi_{\mathbf{P}}}{\mathbf{e}_3^{T\,G}\mathbf{r}_k}A_{\mathbf{P}}\mathbf{C}_k\lfloor{}^G\mathbf{v}_k\times\rfloor\int_{t_1}^{t_k}\mathbf{C}_t^T\mathrm{d}t - B_{\mathbf{P}}. \tag{56}$$

9 Appendix II: The Proof of Theorem I

Proof: To verify the result, we just directly multiplied each block row of $\mathbf{M}(\mathbf{x})$ with $\mathbf{N}$. When

$$k = 1, \quad \mathbf{H}_1\mathbf{N} = \begin{bmatrix} \mathbf{H}_{\delta\mathbf{r},1}\mathbf{N}_1 & -\mathbf{H}_{\delta\mathbf{v},1}\lfloor{}^G\mathbf{v}_1\times\rfloor\mathbf{g} + \mathbf{H}_{\delta\boldsymbol{\theta},1}\mathbf{C}_1\mathbf{g} \end{bmatrix}, \tag{57}$$

for

$$k \geq 2, \quad \mathbf{M}_k\mathbf{N} = \begin{bmatrix} \mathbf{M}_{k1}\mathbf{N}_1 & -\mathbf{M}_{k2}\lfloor{}^G\mathbf{v}_1\times\rfloor\mathbf{g} + \mathbf{M}_{k3}\mathbf{C}_1\mathbf{g} \end{bmatrix}. \tag{58}$$

Since $\mathbf{e}_3^T = \begin{bmatrix} 0 & 0 & 1 \end{bmatrix}$, both $\frac{\mathbf{e}_3^T\mathbf{C}_k^T\pi_{\mathbf{P}}}{\mathbf{e}_3^{T\,G}\mathbf{r}_k{}^G\mathbf{r}_k{}^T\mathbf{e}_3}A_{\mathbf{P}}\mathbf{C}_k{}^G\mathbf{v}_k\mathbf{e}_3^T$ and $\frac{A_{\mathbf{P}}\mathbf{C}_k{}^G\mathbf{v}_k\mathbf{e}_3^T}{\mathbf{e}_3^{T\,G}\mathbf{r}_k}$ had such structure $\begin{bmatrix} 0 & 0 & * \\ 0 & 0 & * \end{bmatrix}$, which were marked by $\boldsymbol{\Sigma}_{\mathbf{k}}^1$ and $\boldsymbol{\Sigma}_{\mathbf{k}}^2$, respectively, hence

$$\mathbf{H}_{\delta\mathbf{r},1}\mathbf{N}_1 = \boldsymbol{\Sigma}_{\mathbf{k}}^1\mathbf{N}_1 = 0_{2\times2}, \quad \mathbf{M}_{k1}\mathbf{N}_1 = \boldsymbol{\Sigma}_{\mathbf{k}}^1\mathbf{N}_1 = 0_{2\times2}, \tag{59}$$

while

$$-\mathbf{H}_{\delta\mathbf{v},1}\lfloor{}^G\mathbf{v}_1\times\rfloor\mathbf{g} + \mathbf{H}_{\delta\boldsymbol{\theta},1}\mathbf{C}_1\mathbf{g} = \frac{A_{\mathbf{P}}\mathbf{C}_1{}^G\mathbf{v}_1\mathbf{e}_3^T}{\mathbf{e}_3^{T\,G}\mathbf{r}_1}\lfloor\mathbf{C}_1^T\pi_{\mathbf{P}}\times\rfloor\mathbf{C}_1^T\mathbf{C}_1\mathbf{g}$$
$$= \boldsymbol{\Sigma}_1^2\lfloor\mathbf{C}_1^T\pi_{\mathbf{P}}\times\rfloor\mathbf{g}, \tag{60}$$

and

$$-\mathbf{M}_{k2}\lfloor{}^G\mathbf{v}_1\times\rfloor\mathbf{g} + \mathbf{M}_{k3}\mathbf{C}_1\mathbf{g} = -\Delta t_k\boldsymbol{\Sigma}_{\mathbf{k}}^1\lfloor{}^G\mathbf{v}_1\times\rfloor\mathbf{g} + \boldsymbol{\Sigma}_{\mathbf{k}}^1$$
$$\cdot \left\lfloor \left({}^G\mathbf{r}_1 + {}^G\mathbf{v}_1\Delta t_k + \frac{1}{2}\mathbf{g}\Delta t_k^2 - {}^G\mathbf{r}_k \right)\times \right\rfloor\mathbf{g}$$
$$- \frac{\mathbf{e}_3^T\mathbf{C}_k^T\pi_{\mathbf{P}}}{\mathbf{e}_3^{T\,G}\mathbf{r}_k}A_{\mathbf{P}}\mathbf{C}_k\cdot\lfloor(\mathbf{g}\Delta t_k)\times\rfloor\mathbf{g} + \boldsymbol{\Sigma}_{\mathbf{k}}^2\lfloor\mathbf{C}_k^T\pi_{\mathbf{P}}\times\rfloor\mathbf{g}. \tag{61}$$

Given $\boldsymbol{\Sigma}_{\mathbf{k}}^1\lfloor\cdot\times\rfloor\mathbf{g} = 0_{2\times1}$, $\boldsymbol{\Sigma}_{\mathbf{k}}^2\lfloor\cdot\times\rfloor\mathbf{g} = 0_{2\times1}$, and $\lfloor\mathbf{g}\times\rfloor\mathbf{g} = 0_{2\times1}$, it yielded $\mathbf{H}_1\mathbf{N} = 0_{2\times3}$, $\mathbf{M}_k\mathbf{N} = 0_{2\times3}$, so that we had $\mathbf{M}(\mathbf{x})\mathbf{N} = 0$.

References

1. L. Schmitt and W. Fichter, "Observability criteria and null-measurement Kalman filter for vision-aided inertial navigation," *J. Guid. Control Dyn.* **39**(4), 770–780 (2016).
2. C. Xu, D. Wang, and X. Huang, "Autonomous navigation based on sequential images for planetary landing in unknown environments," *J. Guid. Control Dyn.* **40**(10), 2587–2602 (2017).
3. Y. Liu and C. Wang, "Hybrid real-time stereo visual odometry for unmanned aerial vehicles," *Opt. Eng.* **57**(7), 073104 (2018).
4. J. Hosen et al., "Vision-aided nonlinear observer for fixed-wing unmanned aerial vehicle navigation," *J. Guid. Control Dyn.* **39**(8), 1777–1789 (2016).
5. X. Zhang et al., "Integrated navigation method based on inertial navigation system and Lidar," *Opt. Eng.* **55**(4), 044102 (2016).
6. R. Mur-Artal, J. M. M. Montiel, and J. D. Tardos, "ORB-SLAM: a versatile and accurate monocular SLAM system," *IEEE Trans. Rob.* **31**(5), 1147–1163 (2015).
7. H. He, Y. Li, and J. Tan, "Relative motion estimation using visual–inertial optical flow," *Auton. Rob.* **42**(3), 615–629 (2018).
8. M. Mammarella et al., "Comparing optical flow algorithms using 6-dof motion of real-world rigid objects," *IEEE Trans. Syst. Man Cybern. Part C* **42**(6), 1752–1762 (2012).
9. F. Kendoul, I. Fantoni, and K. Nonami, "Optic flow-based vision system for autonomous 3D localization and control of small aerial vehicles," *Rob. Autom. Syst.* **57**(6–7), 591–602 (2009).
10. F. Raudies and H. Neumann, "A review and evaluation of methods estimating ego-motion," *Comput. Vision Image Understanding* **116**(5), 606–633 (2012).
11. S. Baker et al., "A database and evaluation methodology for optical flow," *Int. J. Comput. Vision* **92**(1), 1–31 (2011).
12. N. Gageik, M. Strohmeier, and S. Montenegro, "An autonomous UAV with an optical flow sensor for positioning and navigation," *Int. J. Adv. Rob. Syst.* **10**(10), 341 (2013).
13. T. P. Webb et al., "Vision-based state estimation for autonomous micro air vehicles," *J. Guid. Control Dyn.* **30**(3), 816–826 (2007).
14. J. Waldmann, R. I. G. da Silva, and R. A. J. Chagas, "Observability analysis of inertial navigation errors from optical flow subspace constraint," *Inf. Sci.* **327**, 300–326 (2016).
15. J. Sola, "Quaternion kinematics for the error-state Kalman filter," arXiv: 1711.02508, Spanish National Research Council, Barcelona, Spain (2017) not yet published.
16. M. Bloesch et al., "State estimation for legged robots-consistent fusion of leg kinematics and IMU," in *Robotics: Science and Systems VIII*, N. Roy, P. Newman, and S. Srinivasa, Eds., pp. 17–24, The MIT Press, Sydney (2013).
17. K. Fujita et al., "A debris image tracking using optical flow algorithm," *Adv. Space Res.* **49**(5), 1007–1018 (2012).
18. R. W. Keener, "Estimating equations and maximum likelihood," in *Theoretical Statistics: Topics for a Core Course*, G. Casella, S. Fienberg, and I. Olkin, Eds., 1st ed., pp. 175–201, Springer, New York (2010).
19. Y. Pawitan, "Score function and Fisher information," in *In All Likelihood: Statistical Modelling and Inference Using Likelihood*, pp. 213–228, Oxford University Press, Oxford (2001).
20. A. B. Chatfield, "Inertial instrumentation," in *Fundamentals of High Accuracy Inertial Navigation*, P. Zarchan, Ed., Progress in Astronautics and Aeronautics, Vol. **174**, pp. 59–78, AIAA, New York (1997).
21. A. I. Mourikis and S. I. Roumeliotis, "A multi-state constraint Kalman filter for vision-aided inertial navigation," in *IEEE Int. Conf. Rob. and Autom.*, IEEE, pp. 3565–3572 (2007).
22. N. El-Sheimy, H. Hou, and X. Niu, "Analysis and modeling of inertial sensors using Allan variance," *IEEE Trans. Instrum. Meas.* **57**(1), 140–149 (2008).
23. J. Rehder et al., "Extending kalibr: calibrating the extrinsics of multiple IMUs and of individual axes," in *IEEE Int. Conf. Rob. and Autom.*, IEEE, pp. 4304–4311 (2016).
24. D. P. Koch et al., "Relative multiplicative extended Kalman filter for observable GPS-denied navigation," Technical Report (2017), http://scholarsarchive.byu.edu/facpub/1963.
25. J. A. Hesch et al., "Camera-IMU-based localization: observability analysis and consistency improvement," *Int. J. Rob. Res.* **33**(1), 182–201 (2014).
26. J. A. Hesch et al., "Consistency analysis and improvement of vision-aided inertial navigation," *IEEE Trans. Rob.* **30**(1), 158–176 (2014).

Jia Deng received her BS degree at Beihang University, Beijing, in 2010, and she is currently pursuing her PhD at Beihang University, Beijing, China. She mainly studies navigation, control theory, and corresponding application and recently has focused on the data fusion problems and cooperative formation flight control system.

Sentang Wu received his PhD in dynamics, ballistics, and aircraft motion control systems from Ukraine National Aviation University in 1992. He mainly studies the theory and application of nonlinear stochastic systems, computer information processing, and control and aircraft cooperative control. He is currently a professor of automation science and electrical engineering and a doctoral tutor at Beihang University.

Hongbo Zhao received her BS degree at Beihang University, Beijing, and she is currently pursuing her PhD at Beihang University, Beijing, China. She mainly studies the control theory and application, distributed state estimation of stochastic systems, which focus on multiagent systems, and cooperative formation flight control systems with uncertainties.

Da Cai received his BS degree in control science and engineering from Beihang University, Beijing, China, in 2012 and also received his PhD in control science and engineering in 2018. His research interests include multirobot networks, distributed algorithms, and aircraft cooperative control.