Vision-aided Estimation of Attitude, Velocity, and Inertial Measurement Bias for UAV Stabilization

Shiyu Zhao · Feng Lin · Kemao Peng ·
Xiangxu Dong · Ben M. Chen · Tong H. Lee

Abstract This paper studies vision-aided inertial navigation of small-scale unmanned aerial vehicles (UAVs) in GPS-denied environments. The objectives of the navigation system are to firstly online estimate and compensate the unknown inertial measurement biases, secondly provide drift-free velocity and attitude estimates which are crucial for UAV stabilization control, and thirdly give relatively accurate position estimation such that the UAV is able to perform at least a short-term navigation when the GPS signal is not available. For the vision system, we do not presume maps or landmarks of the environment.

S. Zhao (✉) · B. M. Chen · T. H. Lee
Department of Electrical, Computer Engineering, National
University of Singapore, Singapore, Singapore
e-mail: shiyuzhao@nus.edu.sg

B. M. Chen
e-mail: bmchen@nus.edu.sg

T. H. Lee
e-mail: eleleeth@nus.edu.sg

F. Lin · K. Peng · X. Dong
Temasek Laboratories, National University of Singapore,
Singapore, Singapore

F. Lin
e-mail: linfeng@nus.edu.sg

K. Peng
e-mail: kmpeng@nus.edu.sg

X. Dong
e-mail: tsldngx@nus.edu.sg

The vision system should be able to work robustly even given low-resolution images (e.g., 160×120 pixels) of near homogeneous visual features. To achieve these objectives, we propose a novel homography-based vision-aided navigation system that adopts four common sensors: a low-cost inertial measurement unit, a downward-looking monocular camera, a barometer, and a compass. The measurements of the sensors are fused by an extended Kalman filter. Based on both analytical and numerical observability analyses of the navigation system, we theoretically verify that the proposed navigation system is able to achieve the navigation objectives. We also show comprehensive simulation and real flight experimental results to verify the effectiveness and robustness of the proposed navigation system.

1 Introduction

Small-scale unmanned aerial vehicles (UAVs) have attracted a large amount of interests in both academic research and industrial applications in recent years [1, 2]. Due to the payload limitations, small-scale UAVs are widely equipped with light-weight and low-cost inertial measurement units (IMUs) for navigation purposes. The measurements of low-cost IMUs usually

are characterized by high noises and large biases. As a result, pure inertial navigation based on low-cost IMUs drifts rapidly. In practice, inertial navigation is usually aided by the global positioning system (GPS) to achieve drift-free navigation. The GPS signal is, however, unavailable in indoor and certain outdoor environments. Vision techniques can be applied to solve UAV navigation in GPS-denied environments.

Considering there are a large amount of existing studies on vision-aided navigation of UAVs, we first clarify the specific problem considered in this paper. Suppose a small-scale UAV is navigating based on vision and inertial sensors in an indoor or outdoor GPS-denied environment. For the vision system, we do not presume maps or landmarks of the environment. The vision system should be able to work robustly even given low-resolution images of near homogeneous ground scenes such as indoor concrete floor or outdoor grass field. For the inertial sensors, the measurement of the low-cost IMU is corrupted by unknown constant biases. The biases may vary every time the IMU is initialized. As a result, they must be estimated online and then compensated in the navigation algorithm.

The objectives of the navigation system are to estimate and compensate the IMU measurement biases, and concurrently estimate the attitude, velocity, and position of the UAV. The attitude and the velocity must be estimated without drift because they are crucial for the UAV stabilization control. The position estimation is, however, not required to be drift-free as neither known markers nor loop closure of the UAV path is presumed. But the position estimation should be much more accurate than that of the pure inertial navigation such that the UAV is able to perform at least short-term navigation when the GPS signal is not available.

Vision-based navigation of mobile robots has been investigated extensively up to now [3–12]. We refer to [13] for a recent review on this topic. Simultaneous localization and mapping (SLAM) is a popular technique for UAV navigation in unknown environments. SLAM is not employed in our work due to the following reasons. The UAV needs to navigate through an environment with near homogeneous features such as indoor concrete floor or outdoor grass field. In the meantime, the resolution of the image may be very low (e.g., 160×120 pixels) due to the extremely limited onboard computational resources. Since the data association and loop closure in SLAM require high-quality

feature tracking and matching, the above two conditions arise critical challenges for SLAM. In addition, the navigation task considered in this paper requires drift-free attitude and velocity estimation. In order to remove the estimation drift, SLAM usually requires loop closure. But loop closure or any specific path of the UAV is not presumed in our work.

Compared to SLAM, techniques based optical flow usually can work robustly under various conditions such as low-resolution images of near homogeneous features. When the onboard camera is downward-looking and the ground is flat, homography matrices can be calculated from the feature correspondences between consecutive images. Homography has been successfully applied to a variety of vision-based UAV navigation tasks [14–19]. We refer to [20, Section 5.3] for a good introduction to homography. A homography matrix carries certain useful motion information of the UAV. The motion information can be retrieved by decomposing the homography matrix [15, 16]. To avoid homography decomposition, the works in [14, 18] utilized inertial measurements to eliminate the rotation in the homography and then retrieves the translational information. The problem setups in [14–16, 18, 19] are, however, different from ours as the attitude can be directly measured in their works.

Vision-based attitude estimation for UAVs is also a hot research topic [21, 22] in recent years. A recent review on this topic can be found in [22]. The approaches for vision-based UAV attitude estimation include horizon detection, vanishing points, and so on. These approaches, however, are merely designed for attitude estimation. They are not able to simultaneously estimate velocity and IMU measurement biases as our task requires. The work in [23] analyzed the observability of visual and inertial data fusion systems. By assuming multiple point features can be observed during a time interval, it is shown that the attitude, velocity, and IMU measurement biases are observable. The closed-form solutions of these quantities are derived in terms of the visual and inertial measurements. However, it is assumed that multiple point features must be observed for at least five times during the flight. This assumption may not be satisfied in practice. For example, consider a UAV flying forward. If the overlap area between two consecutive images is less than fifty percent of each image, each feature can only be observed at most twice during the flight.

In this paper, we choose optical flow and homography as the fundamental techniques for the vision system as they can robustly handle low-resolution images (e.g., 320×240 or 160×120 pixels) and near homogeneous visual features. The two techniques have been widely applied in vision-aided navigation tasks [14–19]. However, the existing studies only require velocity estimation, whereas the navigation task considered in this paper requires the estimation of velocity, attitude, and IMU measurement biases concurrently. The velocity estimation can be obtained from optical flow and homography measurements as in [14–16, 18, 19], but in order to estimate the attitude and IMU measurement concurrently a new information fusion structure is demanded. To estimate the attitude (specifically, pitch and roll), we adopt the assumption that the ground scene is a level plane and then show that the attitude information is buried in the homography measurements. The assumption of level ground is valid for normal indoor rooms [19]. It is also common to treat the ground as a level plane when the UAV flies at a relatively high altitude in outdoor [15]. The environments that do not satisfy the assumption are not considered in this work.

Homography decomposition is one popular method to retrieve the useful information buried in the homography measurements. However, the homography measurements are computed from the corresponding features between consecutive images. Localization errors or mismatchings of the features would certainly cause errors in the homography. In order to well handle the homography error, we mathematically formulate it in the system modeling in our navigation system. More specifically, we design an extended Kalman filter (EKF) to fuse the measurements from four common sensors including a low-cost IMU, a downward-looking monocular camera, a barometer, and a compass. The homography measurement will be directly input to the EKF without decomposition such that the homography error can be handled by the EKF in a proper manner. Compared to the existing studies [14–19], another contribution of our work is to present both analytical and numerical observability analyses of the proposed navigation system. We show that the attitude, velocity, and IMU biases are all observable when the UAV speed is nonzero. As a result, it is theoretically verified that the proposed navigation system fulfills all the navigation requirements. Moreover, simulation results are shown to verify the effectiveness

of the navigation system. Our comprehensive simulation adopts a nonlinear dynamic model of a real helicopter, a flight control law, and real image processing. Flight experimental results based on a quadrotor UAV are also presented. Both of the simulation and experimental results are consistent with the observability analysis. They successfully verify the effectiveness and robustness of the proposed vision-aided navigation system.

This paper is organized as follows. The vision-aided navigation system is designed in Section 2. Then the observability of the proposed navigation system is analyzed in Section 3. Simulation and real flight experimental results are shown in Sections 4 and 5, respectively. Conclusions are drawn in Section 6.

2 Design of the Vision-aided Navigation System

We first introduce the four types of sensors adopted by the navigation system. The navigation system contains two main sensors: an IMU and a monocular camera. The IMU measures the acceleration (also known as specific force) and angular rate of the UAV. It is assumed that the IMU measurements are corrupted by zero-mean Gaussian white noises and constant biases. Since the biases may vary every time the IMU is initialized, they must be estimated online and then compensated in the navigation algorithm. The monocular camera is directed downward to capture images of the ground scene during flight. The vision measurements, homography matrices, can be computed from consecutive images. We assume each entry of the homography matrix is corrupted by a zero-mean Gaussian white noise. The noises of different entries may have different standard deviations. Furthermore, considering the altitude is crucial for the safety of the UAV, we assume the altitude can be accurately measured by a barometer (or any other altitude sensor). Finally, since it is impossible for vision to estimate the yaw angle without any global references, the yaw angle is directly measured using a compass. Note the yaw angle will generally not affect the stabilization of the UAV even if it may not be measured accurately.

The structure of the navigation system is given in Fig. 1. The measurements of the four sensors are fused by a 15th-order EKF. The 15 states of the EKF are: 3-dimensional (3D) position, 3D velocity, 3D attitude

Fig. 1 The structure of the proposed vision-aided navigation system
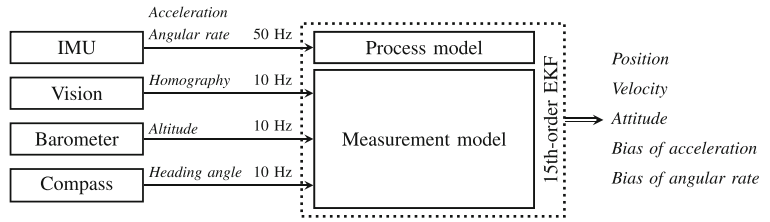
(roll, pitch, and yaw), 3D acceleration bias, and 3D angular rate bias. It is worth noting that the IMU measurement enters the EKF through the process model, whereas the measurements of the vision, barometer, and compass enter the EKF through the measurement model. There exist two update rates in the EKF. The update rate of the process model (i.e., the IMU measurement) is 50 Hz, whereas that of the measurement model (i.e., the measurements of vision, barometer, and compass) is 10 Hz. Denote T_s and T_v as the sampling periods of the processing and the measurement models, respectively. Then $T_s = 0.02$ sec and $T_v = 0.1$ sec.

There are three reference frames in the navigation system: camera frame, body frame, and navigation frame. For the sake of simplicity, the camera is installed on the UAV in the way that the axes of the camera frame are parallel to those of the body frame. The origins of the two frames are very close. As a result, it can be assumed that the camera frame coincides with the body frame. The navigation frame is a local north-east-down frame with its origin located on the ground plane. As the ground is assumed to be a level plane, the x-y plane of the navigation frame coincides with the ground plane. The body and the navigation frames are denoted by subscripts b and n, respectively. We use a slash to represent a transformation from one frame to the other. For example, subscript n/b represents a transformation from the body frame to the navigation frame.

2.1 Process Model

We next design the process model of the navigation system.

Let $\mathbf{p}_n = [p_{n,x}, p_{n,y}, p_{n,z}]^T \in \mathbb{R}^3$ and $\mathbf{v}_n = [v_{n,x}, v_{n,y}, v_{n,z}]^T \in \mathbb{R}^3$ respectively be the position and the velocity of the UAV in the navigation frame. The attitude represented by Euler angles (roll, pitch, and yaw) is denoted as $\rho = [\phi, \theta, \psi]^T \in \mathbb{R}^3$. Denote $\mathbf{e}_i$ with $i \in \{1, 2, 3\}$ as the ith column vector of the 3

by 3 identity matrix $\mathbf{I}_{3\times3}$. The kinematic model of the UAV is

$$\begin{bmatrix} \dot{\mathbf{p}}_n \\ \dot{\mathbf{v}}_n \\ \dot{\rho} \end{bmatrix} = \begin{bmatrix} \mathbf{v}_n \\ \mathbf{R}_{n/b}\,\mathbf{a}_{nb} + g\mathbf{e}_3 \\ \mathbf{L}_{n/b}\,\omega^b_{b/n} \end{bmatrix}, \qquad (1)$$

where $\mathbf{a}_{nb} \in \mathbb{R}^3$ and $\omega^b_{b/n} \in \mathbb{R}^3$ respectively denote the acceleration and angular rate of the UAV in the body frame; and g represents the local gravitational acceleration. The transformation matrices $\mathbf{R}_{n/b}$ and $\mathbf{L}_{n/b}$ are given by

$$\mathbf{R}_{n/b} = \begin{bmatrix} c_\theta c_\psi & s_\phi s_\theta c_\psi - c_\phi s_\psi & c_\phi s_\theta c_\psi + s_\phi s_\psi \\ c_\theta s_\psi & s_\phi s_\theta s_\psi + c_\phi c_\psi & c_\phi s_\theta s_\psi - s_\phi c_\psi \\ -s_\theta & s_\phi c_\theta & c_\phi c_\theta \end{bmatrix},$$

$$\mathbf{L}_{n/b} = \begin{bmatrix} 1 & s_\phi t_\theta & c_\phi t_\theta \\ 0 & c_\phi & -s_\phi \\ 0 & s_\phi/c_\theta & c_\phi/c_\theta \end{bmatrix}, \qquad (2)$$

where $s_* = \sin(*)$, $c_* = \cos(*)$, and $t_* = \tan(*)$.

Let $\mathbf{a}_{nb,\text{IMU}}$ and $\omega^b_{b/n,\text{IMU}}$ be the acceleration and angular rate measured by the IMU, respectively. Then we have

$$\mathbf{a}_{nb,\text{IMU}} = \mathbf{a}_{nb} - \mathbf{b}_a - \mathbf{w}_a, \qquad (3)$$

$$\omega^b_{b/n,\text{IMU}} = \omega^b_{b/n} - \mathbf{b}_\omega - \mathbf{w}_\omega, \qquad (4)$$

where $\mathbf{w}_a \in \mathbb{R}^3$ and $\mathbf{w}_\omega \in \mathbb{R}^3$ are zero-mean Gaussian white noises; and $\mathbf{b}_a = [b_{a,x}, b_{a,y}, b_{a,z}]^T \in \mathbb{R}^3$ and $\mathbf{b}_\omega = [b_{\omega,x}, b_{\omega,y}, b_{\omega,z}]^T \in \mathbb{R}^3$ are unknown but constant measurement biases. Since $\mathbf{b}_a$ and $\mathbf{b}_\omega$ may change every time the IMU is initialized, they must be online estimated and compensated. To that end, the state vector is augmented by adding the unknown biases. From Eqs. 1, 3 and 4, the nonlinear process model of the navigation system is obtained as

$$\begin{bmatrix} \dot{\mathbf{p}}_n \\ \dot{\mathbf{v}}_n \\ \dot{\rho} \\ \dot{\mathbf{b}}_a \\ \dot{\mathbf{b}}_\omega \end{bmatrix} = \begin{bmatrix} \mathbf{v}_n \\ \mathbf{R}_{n/b}(\mathbf{a}_{nb,\text{IMU}} + \mathbf{b}_a + \mathbf{w}_a) + g\mathbf{e}_3 \\ \mathbf{L}_{n/b}(\omega^b_{b/n,\text{IMU}} + \mathbf{b}_\omega + \mathbf{w}_\omega) \\ \mathbf{0}_{3\times1} \\ \mathbf{0}_{3\times1} \end{bmatrix}. \qquad (5)$$

The process model (5) can be rewritten in a compact form as

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, \mathbf{u} + \mathbf{b} + \mathbf{w}), \qquad (6)$$

where

$$\mathbf{x} = \begin{bmatrix} \mathbf{p}_n \\ \mathbf{v}_n \\ \boldsymbol{\rho} \\ \mathbf{b}_a \\ \mathbf{b}_\omega \end{bmatrix}, \quad \mathbf{f}(\mathbf{x}, \mathbf{u} + \mathbf{b} + \mathbf{w}) = \begin{bmatrix} \mathbf{f}_p \\ \mathbf{f}_v \\ \mathbf{f}_\rho \\ \mathbf{f}_{\mathbf{b}_a} \\ \mathbf{f}_{\mathbf{b}_\omega} \end{bmatrix},$$

$$\mathbf{u} = \begin{bmatrix} \mathbf{a}_{\text{nb,IMU}} \\ \boldsymbol{\omega}^b_{\text{b/n, IMU}} \end{bmatrix}, \quad \mathbf{b} = \begin{bmatrix} \mathbf{b}_a \\ \mathbf{b}_\omega \end{bmatrix}, \quad \mathbf{w} = \begin{bmatrix} \mathbf{w}_a \\ \mathbf{w}_\omega \end{bmatrix}. (7)$$

Let

$$\mathbf{A} = \frac{\partial \mathbf{f}}{\partial \mathbf{x}}, \quad \mathbf{B} = \frac{\partial \mathbf{f}}{\partial \mathbf{w}}$$

be the Jacobians of $\mathbf{f}(\mathbf{x}, \mathbf{u} + \mathbf{b} + \mathbf{w})$ with respect to $\mathbf{x}$ and $\mathbf{w}$, respectively. It can be calculated that

$$\mathbf{A} = \begin{bmatrix} \mathbf{0}_{3\times3} & \mathbf{I}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} \\ \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \frac{\partial \mathbf{f}_v}{\partial \rho} & \mathbf{R}_{n/b} & \mathbf{0}_{3\times3} \\ \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \frac{\partial \mathbf{f}_\rho}{\partial \rho} & \mathbf{0}_{3\times3} & \mathbf{L}_{n/b} \\ \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} \\ \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} \end{bmatrix}_{15\times15},$$

$$\mathbf{B} = \begin{bmatrix} \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} \\ \mathbf{R}_{n/b} & \mathbf{0}_{3\times3} \\ \mathbf{0}_{3\times3} & \mathbf{L}_{n/b} \\ \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} \\ \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} \end{bmatrix}_{15\times6}, \qquad (8)$$

where

$$\frac{\partial \mathbf{f}_v}{\partial \boldsymbol{\rho}} = \begin{bmatrix} \frac{\partial \mathbf{R}_{n/b}}{\partial \phi} \mathbf{a}_{\text{nb}} & \frac{\partial \mathbf{R}_{n/b}}{\partial \theta} \mathbf{a}_{\text{nb}} & \frac{\partial \mathbf{R}_{n/b}}{\partial \psi} \mathbf{a}_{\text{nb}} \end{bmatrix}_{3\times3},$$

$$\frac{\partial \mathbf{f}_\rho}{\partial \boldsymbol{\rho}} = \begin{bmatrix} \frac{\partial \mathbf{L}_{n/b}}{\partial \phi} \boldsymbol{\omega}^b_{\text{b/n}} & \frac{\partial \mathbf{L}_{n/b}}{\partial \theta} \boldsymbol{\omega}^b_{\text{b/n}} & \mathbf{0}_{3\times1} \end{bmatrix}_{3\times3}.$$

2.2 Vision Measurement: Homography

Before giving the measurement model of the navigation system, we need first analyze the measurement of the vision system. The onboard camera is directed downward to capture images of the ground scene during flight. The vision measurement, a 3 by 3 homography matrix, can be computed from the matching features of two consecutive images. The detailed algorithm for homography computation can be found in [20, Section 5.3].

Let t and $t_0 = t - T_v$ denote the current and the last sampling time instances, respectively. Given two images captured respectively at t_0 and t, the corresponding features of the two images are related by a homography matrix $\mathbf{H}(t_0, t) \in \mathbb{R}^{3\times3}$. Let $\mathbf{R}(t_0, t) \in \mathbb{R}^{3\times3}$ and $\mathbf{T}(t_0, t) \in \mathbb{R}^3$ respectively be the rotation and translation of the UAV from time t_0 to time t. Denote $\mathbf{N}(t_0) \in \mathbb{R}^3$ as the unit-length normal vector of the ground plane resolved in the camera frame at time t_0. Let $d(t_0) > 0$ be the distance between the UAV and the ground plane at time t_0. Because the ground plane is assumed to be horizontal, the ground plane coincides with the x-y plane of the navigation frame. Thus, $d(t_0)$ is the altitude of the UAV at time t_0. These quantities are all illustrated in Fig. 2. For ease of presentation, the time variables in $\mathbf{H}(t_0, t)$, $\mathbf{R}(t_0, t)$, $\mathbf{T}(t_0, t)$, $\mathbf{N}(t_0)$ and $d(t_0)$ will be omitted in the sequel. Then, the homography $\mathbf{H}$ can be expressed as [20, Section 5.3] [24, Chapter 13]

$$\mathbf{H} = \mathbf{R} + \frac{1}{d} \mathbf{T} \mathbf{N}^T. \qquad (9)$$

It is notable that the terms $\mathbf{R}$, $\mathbf{T}$, $\mathbf{N}$ and d can be further written in the UAV states as

$$\mathbf{R} = \mathbf{R}_{b/n}(t) \mathbf{R}^T_{b/n}(t_0), \qquad (10)$$

$$\mathbf{T} = \mathbf{R}_{b/n}(t) \left[\mathbf{p}_n(t_0) - \mathbf{p}_n(t) \right], \qquad (11)$$

$$\mathbf{N} = \mathbf{R}_{b/n}(t_0) \mathbf{e}_3, \qquad (12)$$

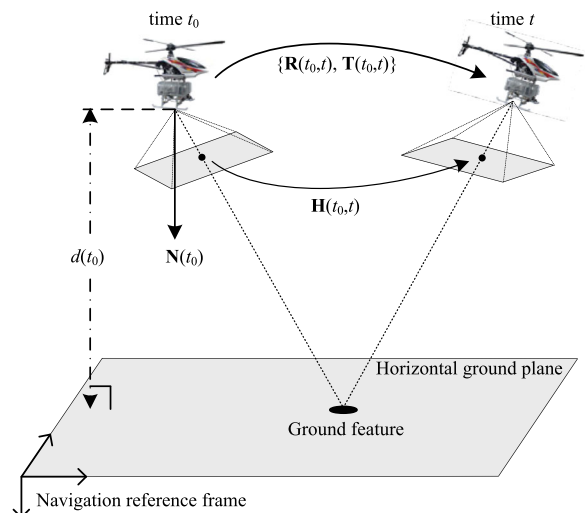$$d = -\mathbf{e}^T_3 \mathbf{p}_n(t_0), \qquad (13)$$

Fig. 2 An illustration of the quantities $\mathbf{R}(t_0, t)$, $\mathbf{T}(t_0, t)$, $\mathbf{N}(t_0)$ and $d(t_0)$ in $\mathbf{H}(t_0, t)$

where $\mathbf{R}_{b/n}$ denotes the rotation from the navigation frame to the body frame and $\mathbf{R}_{b/n} = \mathbf{R}_{n/b}^{T}$. As can be seen, the homography $\mathbf{H}$ clearly is a nonlinear function of the UAV states at both time t and time t_0. It should be noted that Eqs. 12 and 13 are valid only if the ground plane is level. The assumption of level ground is valid for normal indoor rooms and certain outdoor environments. It is also common to treat the ground as a level plane when the UAV flies at a relatively high altitude in outdoor [15].

Since the navigation system is required to estimate the attitude and velocity, we next qualitatively analyze the attitude and velocity information buried in homography.

1) *Attitude Information:* Both $\mathbf{R}$ and $\mathbf{N}$ in Eq. 9 contain certain attitude information of the UAV. The rotation $\mathbf{R}$ represents the incremental attitude of the UAV. It may be useful for estimating the angular rate but useless for drift-free attitude estimation. Recall $\mathbf{N}$ represents the normal vector of the level ground plane resolved in the camera frame. Substituting Eq. 2 into Eq. 12 gives

$$\mathbf{N} = \begin{bmatrix} -s_\theta \\ s_\phi c_\theta \\ c_\phi c_\theta \end{bmatrix}.$$

The above expression clearly shows that $\mathbf{N}$ contains the roll and pitch angles. As a result, we will use $\mathbf{N}$ instead of $\mathbf{R}$ for drift-free attitude estimation in our work. Note the yaw angle is not contained in the homography. Hence it is impossible to obtain drift-free estimation of the yaw angle from the homography. The yaw angle is assumed to be directly measured by a compass, which is a commonly used sensor for UAVs. Unlike the pitch and roll angles, the yaw angle will generally not affect the stabilization of the UAV even if it may not be measured accurately.

2) *Velocity Information:* The term $\mathbf{T}$ in Eq. 9 represents the translation of the UAV from time t_0 to t. But it is expressed in the camera frame according to Eq. 11. We need to transform it from the camera frame to the navigation frame in order to obtain the UAV velocity in the navigation frame. Because the transformation relies on the UAV attitude, drift-free velocity estimation requires drift-free attitude estimation. Another issue to note is the altitude d. According to Eq. 9, the magnitude of $\mathbf{T}$ will be inaccurately scaled if

d is inaccurate. Thus drift-free velocity estimation also requires accurate altitude measurements. Since the altitude is also crucial for the flight safety of the UAV, we assume it can be directly measured by a barometer (or any other altitude sensor).

Up to this point, it is clear that the homography carries useful information of the UAV attitude and velocity. The next problem is how to retrieve the information. One method is to directly decompose $\mathbf{R}$, $\mathbf{T}$ and $\mathbf{N}$ from $\mathbf{H}$ [15, 16]. Interested readers may refer to [20, Section 5.3.3] for homography decomposition algorithms. The decomposition would give two physically possible solutions. Prior knowledge of the UAV motion may be required to disambiguate the two solutions. Homography decomposition can be avoided when inertial measurements are available [14, 18]. As will be shown later, the term $\mathbf{R}$ can be computed accurately using the angular rate measurements. The attitude information in $\mathbf{N}$ and the velocity information in $\mathbf{T}$ can be easily retrieved after eliminating $\mathbf{R}$ from $\mathbf{H}$. But considering that there exist measurement noises in the homography matrix, the homography matrix will be converted to a vector and directly input to the EKF so that the measurement noises can be handled by the EKF in a proper manner.

2.3 Measurement Model

We next design the measurement model of the vision-aided navigation system.

Let $\mathbf{H}$ and $\mathbf{H}_{vis}$ be the true homography and the homography estimated by the vision system, respectively. In our navigation system, we convert $\mathbf{H}_{vis} \in \mathbb{R}^{3 \times 3}$ to $\text{vec}\mathbf{H}_{vis} \in \mathbb{R}^{9}$ and then directly input it into the EKF. The operator vec converts a matrix to a vector by stacking its columns below one another. Suppose each entry of $\mathbf{H}_{vis}$ is corrupted by a zero-mean white noise. Then the nonlinear measurement model for the vision system is

$$\mathbf{y}_{vis} = \text{vec}\mathbf{H}_{vis} = \text{vec}\mathbf{H} + \mathbf{n}_{vis},$$

where $\mathbf{n}_{vis} \in \mathbb{R}^{9}$ is assumed to be a zero-mean Gaussian white noise.

It should be noted that $\text{vec}\mathbf{H}$ is a nonlinear function of both $\mathbf{x}(t)$ and $\mathbf{x}(t_0)$ according to Eqs. 9–13. Since the state vector of the process model is $\mathbf{x}(t)$, we must

express vec$\mathbf{H}$ as a nonlinear function of $\mathbf{x}(t)$ only. Otherwise, the state vector of the EKF must contain both $\mathbf{x}(t)$ and $\mathbf{x}(t_0)$, and the EKF will be not only high-order but also very complicated. To that end, we propose the following approximations to express vec$\mathbf{H}$ as a function of $\mathbf{x}(t)$ only. The approximations may result small errors in the final state estimates, but they would not cause drift in these estimates.

1) *Approximating $\mathbf{R}$:* The term $\mathbf{R}$ in Eq. 10 can be computed from the angular rate measured by the IMU. To see that, the dynamics of $\mathbf{R}_{b/n}$ is given as below:

$$\dot{\mathbf{R}}_{b/n} = -\left[\boldsymbol{\omega}_{b/n}^b\right]_\times \mathbf{R}_{b/n}, \quad (14)$$

where the operator $[\cdot]_\times$ converts a 3D vector to the associated skew-symmetric matrix. The term $\mathbf{R}$ can be obtained by substituting the solution of Eq. 14 into $\mathbf{R} = \mathbf{R}_{b/n}(t)\mathbf{R}_{b/n}^T(t_0)$ as shown in Eq. 10. Equation 14, however, has no closed-form solutions unless $\boldsymbol{\omega}_{b/n}^b$ is (piecewise) constant. Since the sampling period $T_v = 0.1$ sec is short, the angular rate can be treated as constant during the time interval $[t_0, t]$. Then $\mathbf{R}$ can be approximated by

$$\mathbf{R} \approx \exp\left\{-\left[\boldsymbol{\omega}_{b/n}^b(t)\right]_\times T_v\right\}$$
$$\approx \exp\left\{-\left[\boldsymbol{\omega}_{b/n,\,\text{IMU}}^b(t)\right]_\times T_v\right\}. \quad (15)$$

Remark 1 The approximation error of Eq. 15 is caused by firstly the measurement errors in $\boldsymbol{\omega}_{b/n,\,\text{IMU}}^b(t)$ and secondly the assumption that $\boldsymbol{\omega}_{b/n}^b$ is constant during $[t_0, t]$. But because the time interval $T_v = 0.1$ sec is short and the measurement errors in $\boldsymbol{\omega}_{b/n,\,\text{IMU}}^b(t)$ are small, the approximation given by Eq. 15 can be very accurate. This can be verified by numerical simulation results.

Remark 2 The matrix exponential on the right hand side of Eq. 15 can be rigorously computed by Rodrigues' rotation formula [20, Theorem 2.9, p. 27].

1. *Approximating $\mathbf{T}$*: Assume that the UAV velocity is constant during the time interval $[t_0, t]$. Then

we have $\mathbf{p}_n(t) - \mathbf{p}_n(t_0) \approx \mathbf{v}_n(t)T_v$ and hence $\mathbf{T}$ in Eq. 11 can be approximated by

$$\mathbf{T} \approx -\mathbf{R}_{b/n}(t)\mathbf{v}_n(t)T_v. \quad (16)$$

2) *Approximating $\mathbf{N}$*: Recall $\mathbf{N}$ is the normal vector of the ground plane at time t_0. Since $\mathbf{R}_{b/n}(t_0) = \mathbf{R}^T\mathbf{R}_{b/n}(t)$, the vector $\mathbf{N}$ in Eq. 12 can be approximated by

$$\mathbf{N} \approx \mathbf{R}^T\mathbf{R}_{b/n}(t)\mathbf{e}_3, \quad (17)$$

where $\mathbf{R}$ is given by Eq. 15.

3) *Approximating d*: Recall d is the altitude of the UAV at time t_0. Since $\mathbf{p}_n(t) - \mathbf{p}_n(t_0) \approx \mathbf{v}_n(t)T_v$, we can approximately write d in Eq. 13 as

$$d \approx -\mathbf{e}_3^T\left[\mathbf{p}_n(t) - \mathbf{v}_n(t)T_v\right]. \quad (18)$$

Based on the above approximations of $\mathbf{R}$, $\mathbf{T}$, $\mathbf{N}$ and d, we are able to express vec$\mathbf{H}$ as a nonlinear function of $\mathbf{x}(t)$ only. With a little abuse of notation, rewrite the state vector of the EKF as $\mathbf{x} = [x_1, ..., x_{15}]^T$. Then the Jacobian of vec$\mathbf{H}$ with respect to $\mathbf{x}$ is

$$\mathbf{C}_{\text{vis}} = \frac{\partial \text{vec}\mathbf{H}}{\partial \mathbf{x}}$$
$$= \left[\frac{\partial \text{vec}\mathbf{H}}{\partial x_1}, \ldots, \frac{\partial \text{vec}\mathbf{H}}{\partial x_{15}}\right]_{9\times 15}$$
$$= \left[\text{vec}\frac{\partial \mathbf{H}}{\partial x_1}, \ldots, \text{vec}\frac{\partial \mathbf{H}}{\partial x_{15}}\right]_{9\times 15}, \quad (19)$$

where $\partial\mathbf{H}/\partial x_i$ for $i = 1, \ldots, 15$ is given by

$$\frac{\partial \mathbf{H}}{\partial x_i} = -\frac{1}{d^2}\frac{\partial d}{\partial x_i}\mathbf{T}\mathbf{N}^T + \frac{1}{d}\left[\frac{\partial \mathbf{T}}{\partial x_i}\mathbf{N}^T + \mathbf{T}\left(\frac{\partial \mathbf{N}}{\partial x_i}\right)^T\right]. \quad (20)$$

The partial derivatives on the right hand side of Eq. 20 can be calculated from Eqs. 16, 17 and 18.

In addition to the vision measurement model, the measurement models for the compass and the barometer are respectively given by

$$y_{\text{comp}} = \psi + n_{\text{comp}} = \mathbf{C}_{\text{comp}}\mathbf{x} + n_{\text{comp}},$$
$$y_{\text{baro}} = p_{n,z} + n_{\text{baro}} = \mathbf{C}_{\text{baro}}\mathbf{x} + n_{\text{baro}},$$

where $n_{\text{comp}} \in \mathbb{R}$ and $n_{\text{baro}} \in \mathbb{R}$ are assumed to be zero-mean Gaussian white noises and

$$\mathbf{C}_{\text{comp}} = \begin{bmatrix} \mathbf{0}_{1\times 3} & \mathbf{0}_{1\times 3} & \mathbf{e}_3^T & \mathbf{0}_{1\times 3} & \mathbf{0}_{1\times 3} \end{bmatrix}_{1\times 15}, \quad (21)$$
$$\mathbf{C}_{\text{baro}} = \begin{bmatrix} \mathbf{e}_3^T & \mathbf{0}_{1\times 3} & \mathbf{0}_{1\times 3} & \mathbf{0}_{1\times 3} & \mathbf{0}_{1\times 3} \end{bmatrix}_{1\times 15}. \quad (22)$$

To sum up, the nonlinear measurement model of the vision-aided navigation system is

$$\mathbf{y} = \mathbf{h}(\mathbf{x}) + \mathbf{n}, \tag{23}$$

where

$$\mathbf{y} = \begin{bmatrix} \mathbf{y}_{\text{vis}} \\ y_{\text{comp}} \\ y_{\text{baro}} \end{bmatrix}, \quad \mathbf{h}(\mathbf{x}) = \begin{bmatrix} \text{vec}\mathbf{H} \\ \psi \\ p_{\text{n},z} \end{bmatrix}, \quad \mathbf{n} = \begin{bmatrix} \mathbf{n}_{\text{vis}} \\ n_{\text{comp}} \\ n_{\text{baro}} \end{bmatrix}. \tag{24}$$

The Jacobian of $\mathbf{h}(\mathbf{x})$ with respect to $\mathbf{x}$ is given by

$$\mathbf{C} = \frac{\partial \mathbf{h}}{\partial \mathbf{x}} = \begin{bmatrix} \mathbf{C}_{\text{vis}} \\ \mathbf{C}_{\text{comp}} \\ \mathbf{C}_{\text{baro}} \end{bmatrix}_{11 \times 15}. \tag{25}$$

2.4 Extended Kalman Filtering

We have established the continuous process model (6) and the measurement model (23). The corresponding Jacobians have been given in Eqs. 8 and 25. Now it is ready to apply the EKF to fuse the measurements of the IMU, vision, compass, and barometer. The details of the implementation of the EKF are omitted here as they are standard procedures.

In practice, lighting condition changing or insufficient features of the ground scene can cause extremely large homography estimation errors. These inaccurate homography estimates must be detected and rejected. Otherwise, they may cause large errors or even instability of the EKF. Motivated by that, we adopt *innovation filtering* [25, Section 15.3] in our navigation system. Innovation filtering is also called spike filtering or measurement gating. Its basic idea is to compare the real measurement given by sensors with the predicted one given by EKF. If the discrepancy between them exceeds a threshold, then the real measurement for that iteration is rejected. The innovation filter is straightforward to implement and requires little computational resource. Details of innovation filtering are omitted here due to space limitations. Interested readers may refer to [25, Section 15.3].

3 Observability Analysis of the Vision-aided Navigation System

In this section, we present the observability analysis of the proposed navigation system. The purpose of the observability analysis is to identify the observable quantities and theoretically verify if the proposed navigation system is able to fulfill all the requirements.

Since process model (6) and measurement model (23) are highly nonlinear, we only consider the observability of linearized systems. In particularly, we consider two representative linearization conditions: hovering and straight and steady level (SSL) flight. Note hovering is a flight mode only possible for rotorcrafts such as quadrotor UAVs. As will be shown later the observability analysis based on the linearized systems is consistent with the numerical simulation and experimental experiments.

When the UAV is in SSL flight or hovering condition, the UAV states are approximately given by

$$\begin{aligned} \phi = \theta = \psi &= 0, \\ \boldsymbol{\omega}_{\text{b/n}}^{\text{b}} &= \mathbf{0}_{3 \times 1}, \\ \mathbf{a}_{\text{nb}} &= -g\mathbf{e}_3, \\ \mathbf{v}_{\text{n}} &= \kappa\mathbf{e}_1, \end{aligned} \tag{26}$$

where $\kappa \geq 0$ represents the UAV speed. When $\kappa = 0$, the UAV is hovering. As will be shown later, the value of the UAV speed κ can affect the observability of the system. Due to symmetry, the value of the yaw angle has no influence on the observability analysis. For the sake of simplicity, we choose $\psi = 0$.

Substituting condition Eq. 26 into Eq. 8 gives

$$\mathbf{A} = \begin{bmatrix} \mathbf{0}_{3\times3} & \mathbf{I}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} \\ \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & g[\mathbf{e}_3]_\times & \mathbf{I}_{3\times3} & \mathbf{0}_{3\times3} \\ \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{I}_{3\times3} \\ \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} \\ \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} \end{bmatrix}.$$

Substituting condition Eq. 26 into Eqs. 15, 16 and 17 yields $\mathbf{R} = \mathbf{I}_{3\times3}$, $\mathbf{T} = -T_{\text{v}}\mathbf{v}_{\text{n}}$, and $\mathbf{N} = \mathbf{e}_3$, respectively. Further substituting these values into Eq. 19 gives

$$\mathbf{C}_{\text{vis}} = \frac{1}{\alpha} \begin{bmatrix} \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & -\kappa\mathbf{e}_1\mathbf{e}_2^{\text{T}} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} \\ \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \kappa\mathbf{e}_1\mathbf{e}_1^{\text{T}} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} \\ \mathbf{0}_{3\times3} & \mathbf{I}_{3\times3} & \kappa[\mathbf{e}_1]_\times & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} \end{bmatrix}. \tag{27}$$

where $\alpha = -d/T_{\text{v}}$. While calculating (27), we omit the small terms containing T_{v}^2/d^2 or T_{v}/d^2 considering T_{v} is small. Recall $\mathbf{C}_{\text{comp}}$ and $\mathbf{C}_{\text{baro}}$ are given in Eqs. 21 and 22. Then $\mathbf{C}$ can be obtained by

substituting Eqs. 21, 22 and 27 into Eq. 25. The observability matrix is given by

$$\mathcal{O} = \left[\mathbf{C}^{\mathrm{T}}, (\mathbf{CA})^{\mathrm{T}}, \cdots, (\mathbf{CA}^{14})^{\mathrm{T}} \right]^{\mathrm{T}}.$$

We next analyze the rank of the observability matrix under condition (26).

3.1 Case 1: SSL Flight

We first consider the SSL flight condition with $\kappa > 0$. It is easy to obtain the observability matrix in this case as

$$\mathcal{O}_{\mathrm{SSL}} = \left[\begin{array}{ccccc} \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & -\kappa\mathbf{e}_1\mathbf{e}_2^{\mathrm{T}} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} \\ \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \kappa\mathbf{e}_1\mathbf{e}_1^{\mathrm{T}} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} \\ \mathbf{0}_{3\times3} & \mathbf{I}_{3\times3} & \kappa[\mathbf{e}_1]_{\times} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} \\ \mathbf{0}_{1\times3} & \mathbf{0}_{1\times3} & \alpha\mathbf{e}_3^{\mathrm{T}} & \mathbf{0}_{1\times3} & \mathbf{0}_{1\times3} \\ \alpha\mathbf{e}_3^{\mathrm{T}} & \mathbf{0}_{1\times3} & \mathbf{0}_{1\times3} & \mathbf{0}_{1\times3} & \mathbf{0}_{1\times3} \\ \hline \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & -\kappa\mathbf{e}_1\mathbf{e}_2^{\mathrm{T}} \\ \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \kappa\mathbf{e}_1\mathbf{e}_1^{\mathrm{T}} \\ \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & g[\mathbf{e}_3]_{\times} & \mathbf{I}_{3\times3} & \kappa[\mathbf{e}_1]_{\times} \\ \mathbf{0}_{1\times3} & \mathbf{0}_{1\times3} & \mathbf{0}_{1\times3} & \mathbf{0}_{1\times3} & \alpha\mathbf{e}_3^{\mathrm{T}} \\ \mathbf{0}_{1\times3} & \alpha\mathbf{e}_3^{\mathrm{T}} & \mathbf{0}_{1\times3} & \mathbf{0}_{1\times3} & \mathbf{0}_{1\times3} \\ \hline \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & g[\mathbf{e}_3]_{\times} \\ \mathbf{0}_{1\times3} & \mathbf{0}_{1\times3} & \mathbf{0}_{1\times3} & \alpha\mathbf{e}_3^{\mathrm{T}} & \mathbf{0}_{1\times3} \end{array} \right]_{26\times15}.$$

(28)

The scale factor $1/\alpha$ and zero rows of $\mathcal{O}_{\mathrm{SSL}}$ are omitted as they do not contribute to the rank of $\mathcal{O}_{\mathrm{SSL}}$. By examining the row rank, it is straightforward to see

$$\mathrm{rank}(\mathcal{O}_{\mathrm{SSL}}) = 13.$$

Hence there are *two* unobservable modes. In order to identify the two unobservable modes, we need to determine the unobservable subspace (i.e., the null space of $\mathcal{O}_{\mathrm{SSL}}$). By observation, we obtain an orthogonal basis of the unobservable subspace as

$$\mathrm{Null}(\mathcal{O}_{\mathrm{SSL}}) = \mathrm{Range} \left[\begin{array}{cc} \mathbf{e}_1 & \mathbf{e}_2 \\ \mathbf{0}_{3\times1} & \mathbf{0}_{3\times1} \\ \mathbf{0}_{3\times1} & \mathbf{0}_{3\times1} \\ \mathbf{0}_{3\times1} & \mathbf{0}_{3\times1} \\ \mathbf{0}_{3\times1} & \mathbf{0}_{3\times1} \end{array} \right]_{15\times2}.$$

The above equation suggests that the two unobservable modes are

$$\mathbf{x}_{\mathrm{SSL}}^{\mathrm{unobs}} = \{p_{\mathrm{n},x}, \, p_{\mathrm{n},y}\}.$$

Based on the above analysis, we have the following conclusion. In the case of SSL flight, the position ($p_{\mathrm{n},x}$ and $p_{\mathrm{n},y}$) is unobservable, but the velocity, attitude, and IMU biases are all observable. In fact, this is the best situation we can have because it is impossible to make the position observable without any global references such as maps.

3.2 Case 2: Hovering

We now consider the hovering condition with $\kappa = 0$. By substituting $\kappa = 0$ into Eq. 28, $\mathcal{O}_{\mathrm{SSL}}$ degenerates to

$$\mathcal{O}_{\mathrm{hover}} = \left[\begin{array}{ccccc} \mathbf{0}_{3\times3} & \mathbf{I}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} \\ \mathbf{0}_{1\times3} & \mathbf{0}_{1\times3} & \alpha\mathbf{e}_3^{\mathrm{T}} & \mathbf{0}_{1\times3} & \mathbf{0}_{1\times3} \\ \alpha\mathbf{e}_3^{\mathrm{T}} & \mathbf{0}_{1\times3} & \mathbf{0}_{1\times3} & \mathbf{0}_{1\times3} & \mathbf{0}_{1\times3} \\ \hline \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & g[\mathbf{e}_3]_{\times} & \mathbf{I}_{3\times3} & \mathbf{0}_{3\times3} \\ \mathbf{0}_{1\times3} & \mathbf{0}_{1\times3} & \mathbf{0}_{1\times3} & \mathbf{0}_{1\times3} & \alpha\mathbf{e}_3^{\mathrm{T}} \\ \mathbf{0}_{1\times3} & \alpha\mathbf{e}_3^{\mathrm{T}} & \mathbf{0}_{1\times3} & \mathbf{0}_{1\times3} & \mathbf{0}_{1\times3} \\ \hline \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & g[\mathbf{e}_3]_{\times} \\ \mathbf{0}_{1\times3} & \mathbf{0}_{1\times3} & \mathbf{0}_{1\times3} & \alpha\mathbf{e}_3^{\mathrm{T}} & \mathbf{0}_{1\times3} \end{array} \right]_{14\times15}.$$

Note $\mathcal{O}_{\mathrm{hover}}$ is merely a special case of $\mathcal{O}_{\mathrm{SSL}}$. By counting the row rank, it is straightforward to see

$$\mathrm{rank}(\mathcal{O}_{\mathrm{hover}}) = 11.$$

Hence there are *four* unobservable modes. Clearly the observability of the system degenerates when the UAV speed is zero. By observation, we can identify an orthogonal basis of the null space of $\mathcal{O}_{\mathrm{hover}}$ as

$$\mathrm{Null}(\mathcal{O}_{\mathrm{hover}}) = \mathrm{Range} \left[\begin{array}{cccc} \mathbf{e}_1 & \mathbf{e}_2 & \mathbf{0}_{3\times1} & \mathbf{0}_{3\times1} \\ \mathbf{0}_{3\times1} & \mathbf{0}_{3\times1} & \mathbf{0}_{3\times1} & \mathbf{0}_{3\times1} \\ \mathbf{0}_{3\times1} & \mathbf{0}_{3\times1} & \mathbf{e}_1 & \mathbf{e}_2 \\ \mathbf{0}_{3\times1} & \mathbf{0}_{3\times1} & -g\mathbf{e}_2 & g\mathbf{e}_1 \\ \mathbf{0}_{3\times1} & \mathbf{0}_{3\times1} & \mathbf{0}_{3\times1} & \mathbf{0}_{3\times1} \end{array} \right]_{15\times4},$$

from which the unobservable modes can be determined as

$$\mathbf{x}_{\mathrm{hover}}^{\mathrm{unobs}} = \{p_{\mathrm{n},x}, \, p_{\mathrm{n},y}, \, \phi - gb_{a,y}, \, \theta + gb_{a,x}\}.$$

Therefore, the position ($p_{\mathrm{n},x}$ and $p_{\mathrm{n},y}$) is still unobservable in the case of hovering. Moreover, $\phi - gb_{a,y}$ and $\theta + gb_{a,x}$ are also unobservable. In other words, ϕ and θ as well as $b_{a,x}$ and $b_{a,y}$ become unobservable when the UAV speed is zero. This observation can be well explained by using the homography formula (9): the attitude information of the pitch and roll angles originally comes from the normal vector $\mathbf{N}$ in the homography; when the UAV speed is zero, the translational vector $\mathbf{T}$ in Eq. 9 is zero; consequently the term $\mathbf{TN}^{\mathrm{T}}$ will vanish and the attitude information carried by $\mathbf{N}$ cannot be retrieved. Additionally, it can

also be seen from Eq. 27 that the vision measurement degenerates to the velocity measurement when $\kappa = 0$.

3.3 Numerical Rank Analysis

In the preceding subsections, we analytically determined the rank of the observability matrix. In fact, we can also numerically analyze the observability by examining the singular values of the observability matrix. The numerical analysis can provide us new insights into the observability of the navigation system.

Denote the singular values of $\mathcal{O}_{SSL}$ as $\sigma_1 \geq \cdots \geq \sigma_{15} \geq 0$. When $\kappa > 0$ we have rank$(\mathcal{O}_{SSL}) = 13$ and hence $\sigma_{12} \geq \sigma_{13} > 0 = \sigma_{14} = \sigma_{15}$; when $\kappa = 0$ we have rank$(\mathcal{O}_{hover}) = 11$ and hence $\sigma_{12} = \sigma_{13} = \sigma_{14} = \sigma_{15} = 0$. Clearly if $\kappa \to 0$ we would have $\sigma_{12} \to 0$ and $\sigma_{13} \to 0$. Thus it is reasonable to choose the ratio σ_1/σ_{13} as an indicator to evaluate the rank of $\mathcal{O}_{SSL}$: if σ_1/σ_{13} is very large, $\mathcal{O}_{SSL}$ is on the verge of rank deficiency, and the rank of $\mathcal{O}_{SSL}$ almost degenerates to that of $\mathcal{O}_{hover}$. Therefore, if σ_1/σ_{13} is very large, the observability of θ and ϕ would become *weak* though they are still observable. The term weak as used here intuitively means that the estimation of pitch and roll angles may converge slow.

The UAV speed κ is not the only parameter that affects σ_1/σ_{13}. In addition to κ, $\mathcal{O}_{SSL}$ is also parameterized by the UAV altitude d.[1] Figure 3 shows how κ and d affect σ_1/σ_{13}. It is observed from Fig. 3 that σ_1/σ_{13} is large when κ is small or d is large. In other words, the observability of θ and ϕ is weak when the UAV speed is slow or the altitude is high. This observation can be well explained by the homography formula (9): when the speed is slow or the altitude is large, the term $\mathbf{TN}^{\mathrm{T}}/d$ would almost vanish, which will cause difficulty to recover the attitude information in $\mathbf{N}$. This observation can also be explained from a more fundamental point of view, which involves the bearing-only property of vision systems. Note monocular cameras are inherently bearing-only sensors because the depth information of the scene is lost during perspective projection when forming an image. An image inherently only carries the bearings of the ground features. A homography is computed from two

---

[1]The altitude d is contained in $\alpha = -d/T_v$. The vision sampling period T_v can also affect σ_1/σ_{13}. Here we only consider $T_v = 0.1$ sec.
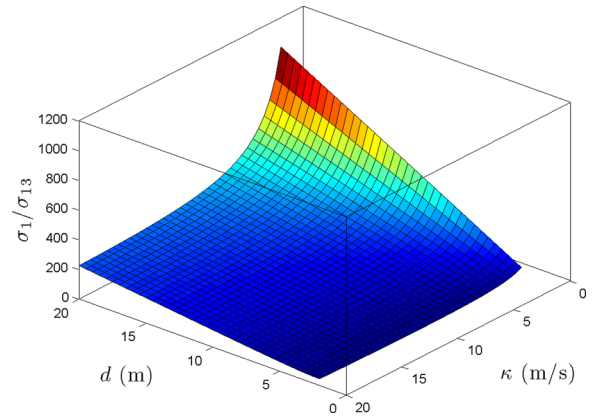

Fig. 3 The ratio σ_1/σ_{13} is large when κ is small or d is large

consecutive images. If the two images are captured from very different angles relative to the ground features, certain useful information (pitch and roll angles in our case) can be recovered from the bearings carried by the two images. Otherwise, if the UAV speed is slow or the UAV altitude is large, the two consecutive images are almost captured from the same angle relative to the ground features. Then the two images would not provide much new information other than bearings. Hence the pitch and roll angles become difficult to estimate. One may refer to [26–29] on the interesting properties of bearing-only measurements.

4 Numerical Simulation

In this section, we show comprehensive simulation results to verify the effectiveness of the proposed vision-aided navigation system.

4.1 Simulation Setup

The structure of our simulation program is shown in Fig. 4. The simulation adopts a 6-DOF nonlinear unmanned helicopter model and a flight control law, the details of which can be found in [1, 30]. Given a trajectory reference, we are able to compute the true states of the UAV. Then we add noises and biases to the true states to generate the simulated measurements of the sensors. Table 1 shows the values of the biases and the standard deviations of the noises.

In our simulation, the homography matrices are computed from partially synthetic images. We have

Fig. 4 Block diagram of the simulation
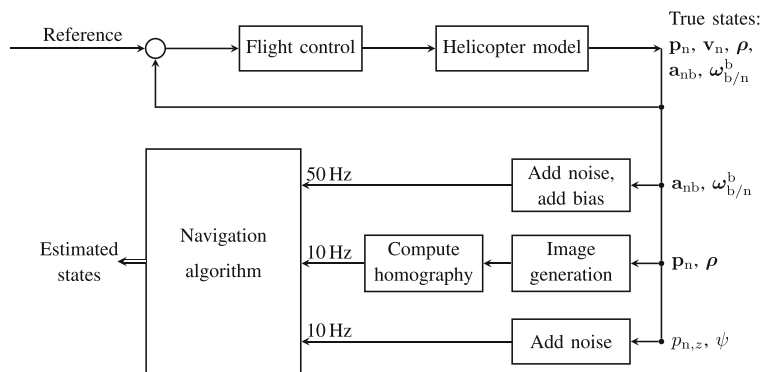
developed a program using Matlab and C++ to generate these synthetic images. In particular, a satellite image (3384×2766 pixels) is used to simulate the ground scene. By setting appropriate intrinsic parameters of the downward-looking camera, the small images (320×240 pixels) of the ground scene are generated according to the position and attitude of the UAV. The small images are used to simulate the images captured by the onboard camera. Samples of the generated images are given in Fig. 5.

We employ OpenCV 2.3.1 in our work to realize real-time vision processing. The following is the procedure to compute the homography from two consecutive images: 1) extracting feature points in each image (OpenCV function *goodFeaturesTo-Track*), 2) matching feature points of the two images (OpenCV function *calcOpticalFlowPyrLK*), and 3) computing homography from feature correspondences (OpenCV function *findHomography*). Note the second largest singular value of a homography equals one [20, Lemma 5.18, p. 135]. Hence the homography given by function *findHomography* should be normalized before usage. In order to improve the homography estimation accuracy, we may utilize a number of auxiliary functions such as histogram equalization (OpenCV function *equalizeHist*) and sub-pixel

feature location refinement (OpenCV function *corner-SubPix*).

4.2 Simulation Results

The trajectory reference of the UAV in the simulation is a sinusoidal wave with constant yaw angle and altitude. This is a typical maneuvering flight motion called *slalom* [1]. Figure 5 shows samples of the generated images. Homography matrices are computed from each pair of consecutive images. Since the true states of the UAV are known in the simulation, true homography matrices can also be computed. Thus we can obtain the errors of the homography estimated by vision (see Fig. 6). As can be seen, the error of each entry of the homography is zero-mean and can be approximately assumed as a zero-mean Gaussian white noise. Most of the homography estimates are accurate though there exist a number of spike measurements whose errors are much larger than the others. In practice, these spikes can be effectively eliminated by innovation filtering.

The UAV states are shown in Fig. 7. The true UAV states are indicated by the green solid lines; the estimated UAV states are indicated by the red dotted lines; and the ones estimated by pure inertial navigation are

Table 1 Noise standard deviation and biases in the simulation

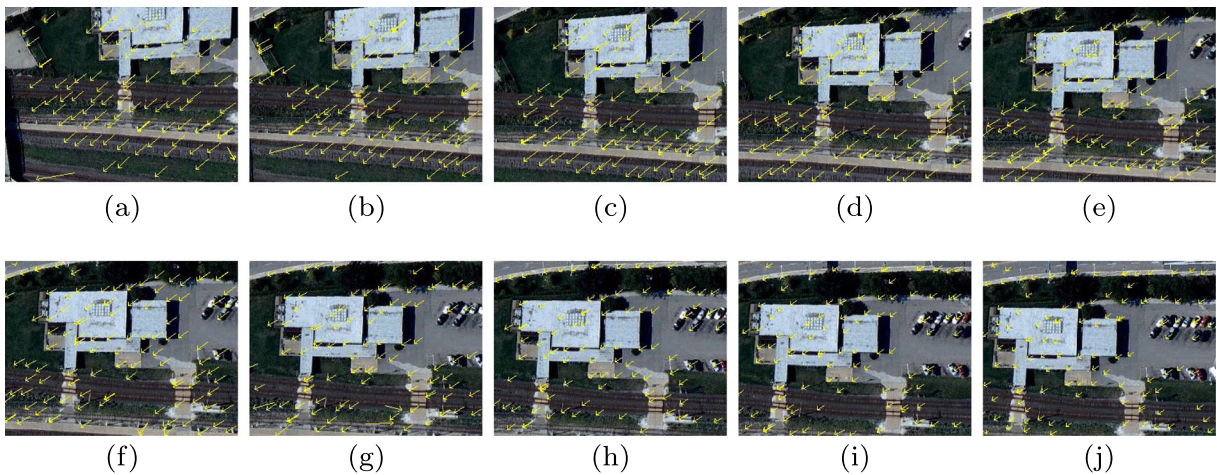| Measurement	Noise standard deviation	Bias
Acceleration (m/s^2)	0.05 (for each entry)	$0.03g$ (for each entry)
Angular rate (rad/s)	0.02 (for each entry)	$\pi/180$ (for each entry)
Yaw angle (degree)	1	None
Altitude (m)	2	None

Fig. 5 Samples of the generated images in the simulation. The size of each image is 320×240 pixels. The *arrows* in the images represent the detected optical flow

indicated by the cyan dashed lines. From Fig. 7, we obtain the following important observations.

1) Due to the large biases in the acceleration and angular rate measurements, the UAV states estimated by pure inertial navigation drift very fast.
2) The attitude and the velocity can be estimated by the vision-aided navigation system accurately without drift as shown in Fig. 7c-d.
3) The biases of the IMU measurements can be estimated accurately as shown in Fig. 7e-f.

4) Vision-aided navigation can significantly reduce the position drift as shown in Fig. 7a-b though the position ($p_{n,x}$ and $p_{n,y}$) estimate still drifts slowly.

The above observations are consistent with our observability analysis in Section 3.1. In addition, we have also conducted simulation under other flight conditions. These results are omitted here due to space limitations. It is observed from these simulation results that the convergence of the estimates of θ, ψ,
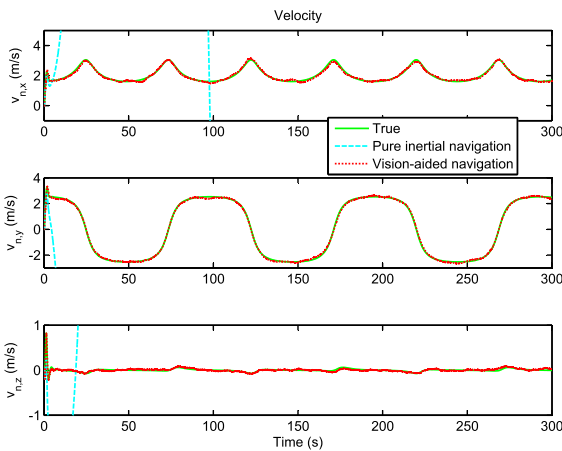
Fig. 6 The homography error in the simulation
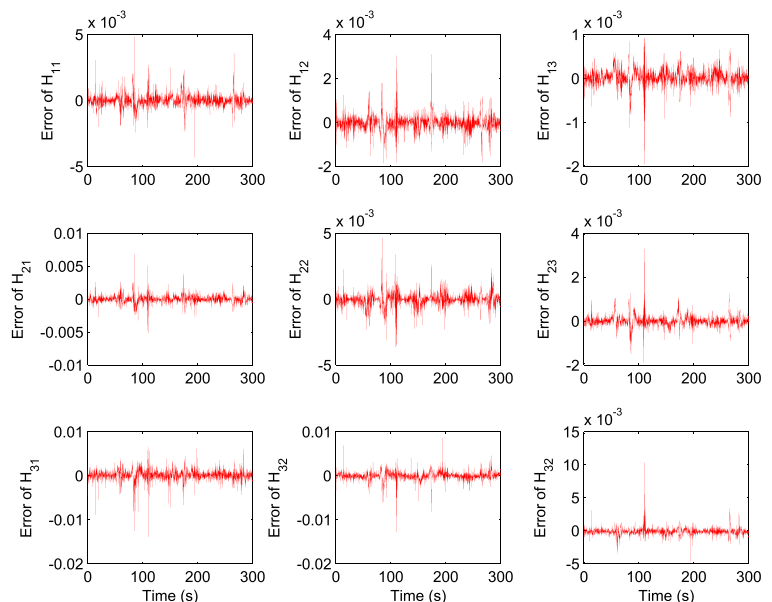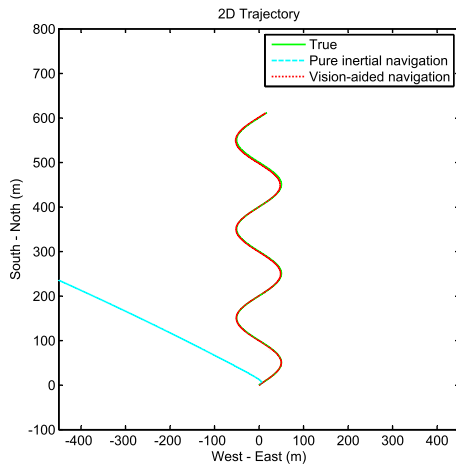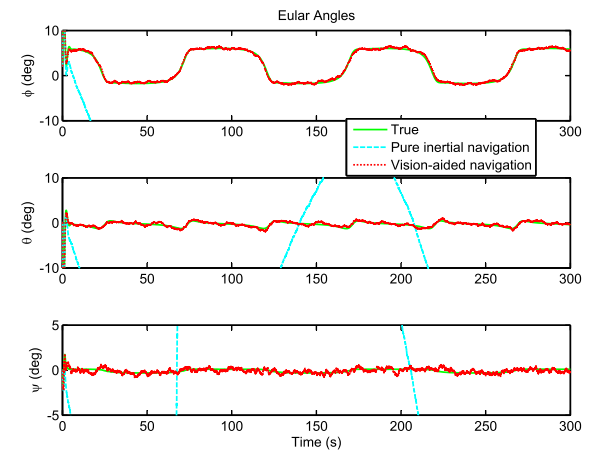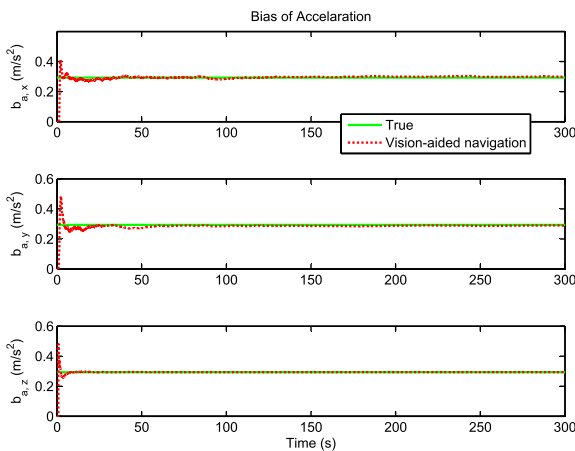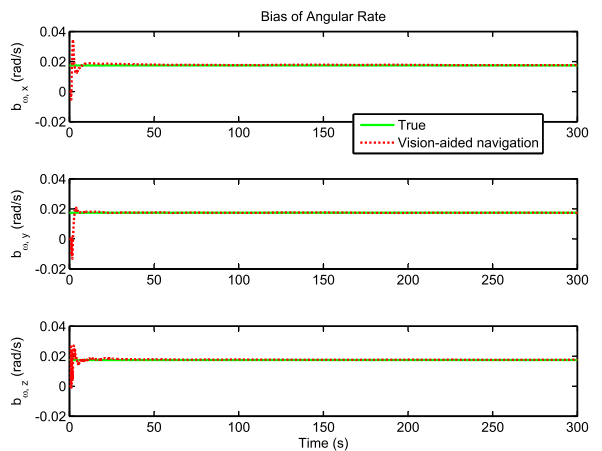
(a) 2D trajectory
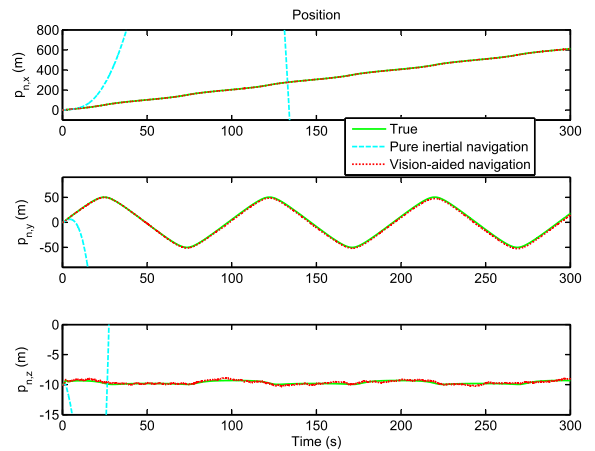
(b) Position

(c) Velocity

(d) Euler angle

(e) Bias of acceleration

(f) Bias of angular rate

Fig. 7 Simulation results

$b_{a,x}$ and $b_{a,y}$ would be slow when the UAV altitude is high or the UAV speed is slow. This observation is also consistent with the numerical observability analysis in Section 3.3.

5 Flight Experiments

In this section, we present flight experimental results to verify the effectiveness and robustness of the proposed navigation system.

5.1 Experimental Platform

The flight experimental platform is an autonomous quadrotor UAV (see Fig. 8) constructed by the NUS UAS group. The key specifications of the quadrotor UAV are listed in Table 2. The structure of the onboard systems is given in Fig. 9. The main onboard systems of this platform are introduced as below.

5.1.1 Navigation Sensors

The quadrotor UAV is equipped with a navigation sensor, IG-500N, which contains an IMU, a GPS receiver and a barometer. The navigation sensor can provide a variety of measurements such as acceleration and angular rate. It can also give drift-free

Fig. 8 The quadrotor UAV and the flat grass field for flight experiments

Table 2 Key specifications of the quadrotor UAV

Specifications	Quadrotor UAV
Dimensions	$61 \times 61 \times 35$ cm
No-load weight	1 kg
Maximum takeoff weight	3 kg
Power source	Lithium polymer battery
Flight endurance	15 mins

measurements of the UAV states in the presence of GPS. The acceleration, angular rate, yaw angle, and altitude given by the navigation sensor are treated as measurements for the vision-aided navigation system. The update rate of the acceleration and angular rate is 50 Hz, while that of the yaw angle and altitude is 10 Hz. The UAV states given by the navigation sensor are treated as the *ground truth* in the flight experiments.

The gyro data used in the experiment has been filtered by the IMU with temperature and gyro-G effect compensation. The bias of the gyro used in our experiment is very small but not zero, and it is around 0.00175 rad/s according to the manual of the sensor. The barometer is fully calibrated and temperature compensated. It is fused with IMU to obtain smooth and high update rate output. Although the well calibrated barometer still has bias which will theoretically cause errors in the velocity estimation, the bias is small compared to the altitude of the UAV and will not affect the system performance significantly. As a result, the small bias in the barometer measurement is simply ignored in the experiment.
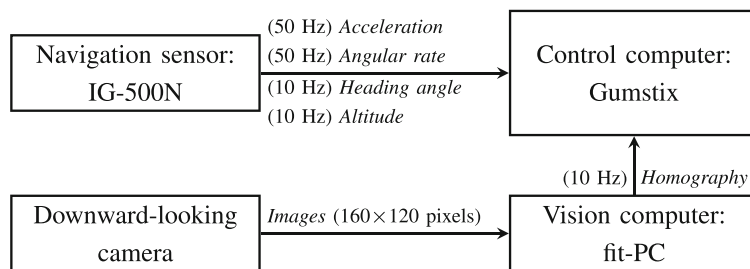
5.1.2 Flight Control System

The primary tasks of the flight control system include collecting measurements from various sensors, executing the proposed navigation algorithm and performing flight control. The flight control system also communicates with the ground control station for real-time monitoring and command issuing. The flight control computer is a Gumstix Overo Fire embedded computer with a 720 MHz processor. The navigation and control algorithms run at 50 Hz in the flight control computer. In order to improve the

Fig. 9 The structure of the onboard systems. The 15th-order EKF is executed in real-time in the control computer

| Navigation sensor: IG-500N | → (50 Hz) *Acceleration* (50 Hz) *Angular rate* (10 Hz) *Heading angle* (10 Hz) *Altitude* → | Control computer: Gumstix |

(10 Hz) *Homography* ↑

| Downward-looking camera | *Images* (160×120 pixels) → | Vision computer: fit-PC |

real-time performance, the original Linux operating system in the Gumstix is replaced by a QNX Neutrino real-time operating system. For details of the onboard software system and the ground control station, please refer to [31, 32]. The quadrotor platform is able to perform autonomous hovering and way point following. For details of the modeling and control of the quadrotor UAV, please refer to [19].

5.1.3 Vision System

The onboard vision system consists of a monocular camera and an embedded vision computer. The weight of the entire vision system is about 350 g. The camera (Pointgrey Firefly) is directed downwards to capture images of the ground scene during flight. It captures images of 160×120 pixels at 10 frame per second. The parameters of the camera such as exposure and shutter speed can be customized for outdoor applications. A wide-angle lens is connected to the camera to enhance the image quality. The intrinsic parameters of the camera have been calibrated before flight experiments. To process the images captured by the camera, a light-weight embedded computer (fit-PC2i)

is adopted as the vision computer. The vision computer contains an Intel Atom Z530 1.6 GHz CPU, a solid-state drive (SSD), 1 GB memory, and four USB 2.0 high-speed ports. Compared to hard disk drives, the SSD is less susceptible to physical vibration during flight. The image and vision processing algorithms are implemented in a Linux operating system. The camera and the vision computer are connected through a USB 2.0 port. Homography matrices are computed in the vision computer at 10 Hz and sent to the flight control computer through a RS232 full UART.

The vision algorithm for computing homography has already been discussed in the last paragraph of Section 4.1. But since the onboard computational resource is very limited, we need to adjust the vision algorithm to realize real-time onboard vision processing. For example, the image size in simulations is 320×240 pixels, but we need to reduce the image size to 160×120 pixels in the flight experiments. Some time-consuming functions such as sub-pixel feature location refinement must also be removed. Doing this can accelerate the vision algorithms but also scarify the accuracy of the estimated homography.

Fig. 10 Samples of the images captured by the onboard camera in the flight experiment. The size of each image is 160×120 pixels. The *arrows* in the images represent the detected optical flow
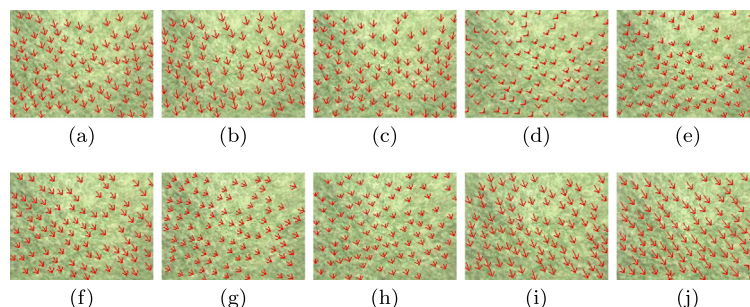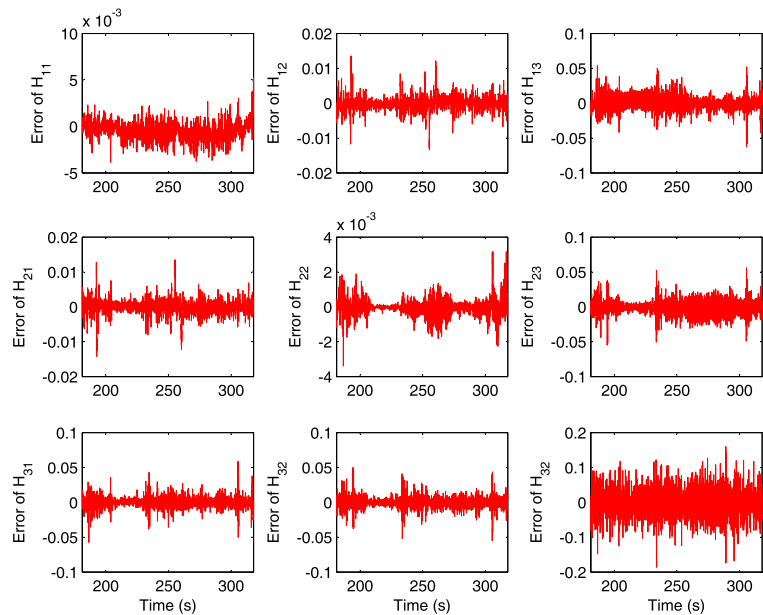

(a) (b) (c) (d) (e)

(f) (g) (h) (i) (j)

5.2 Experimental Results

The flight experiments are conducted in a flat grass field (see Fig. 8). Samples of the images of the grass field captured by the onboard camera are shown in Fig. 10. As can be seen, the quality of the visual features is low. But feature correspondences between each pair of consecutive images can still be smoothly detected based on optical flow.

In the flight experiment, the UAV flies autonomously to follow the pre-specified way points. The UAV states estimated by the vision-aided navigation system are utilized for autonomous flight control. According to the position and attitude provided by GPS/IMU, we can compute the true homography matrices, based on which the errors of the onboard estimated homography can be obtained (see Fig. 11). As can be seen, the error of each entry of the homography can be approximately assumed as a zero-mean Gaussian white noise. The homography errors in the flight experiment are much larger than those in the simulation. But the vision-aided navigation system still performs well.

The closed-loop flight experimental results are shown in Fig. 12. We obtain the following conclusions from the experimental results.

1) The vision-aided navigation can provide accurate and drift-free estimation of the velocity and attitude. The UAV can be successfully stabilized based on the vision-aided navigation.

2) The ground truth for the IMU biases are not available. But it can be implied that these biases are accurately estimated. Note the estimated biases are compensated in the navigation algorithm. If the estimates of the biases are inaccurate, the position, velocity, and attitude given by the vision-aided navigation system will drift rapidly.

3) The position estimate drifts slowly.

The above observations are consistent with the observability analysis in Section 3 and the simulation results in Section 4.

Several remarks on the experimental results are given here. First, the 2D trajectory is plotted against the satellite image of the flight experimental field in Fig. 12a. Note the satellite image is only used for illustration but *not* used for navigation. Second, the z-axis velocity $v_{n,z}$ given by the vision-aided navigation is inconsistent with the GPS data as shown in Fig. 12c. In fact, the vision-aided navigation correctly estimates $v_{n,z}$ because the $p_{n,z}$ can be directly measured by the barometer. The inconsistency is caused by certain technical problems which will be solved in the future.
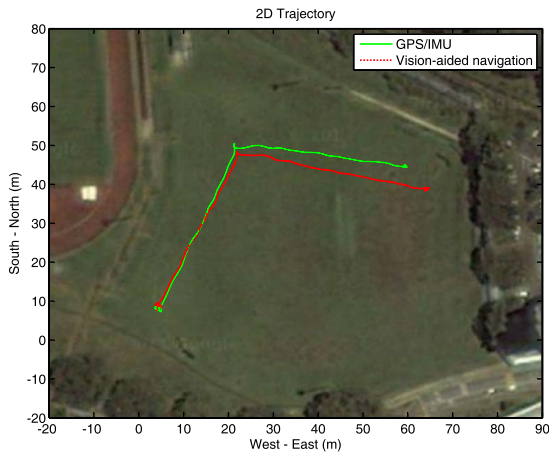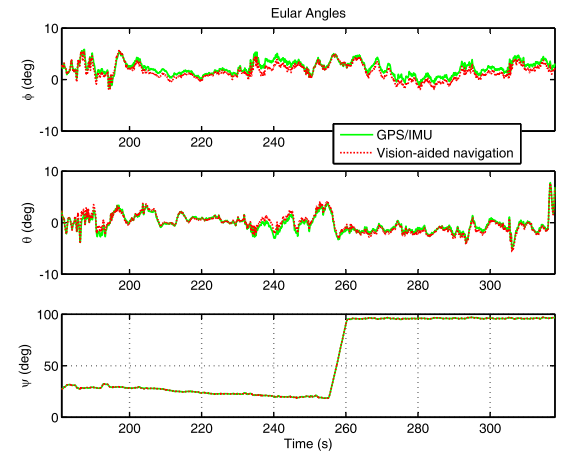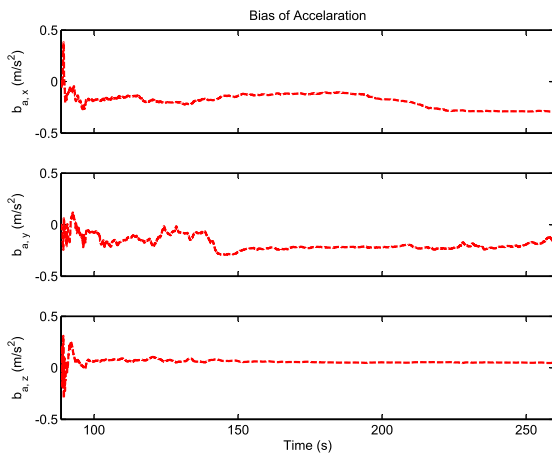
(a) 2D trajectory plotted against satellite image
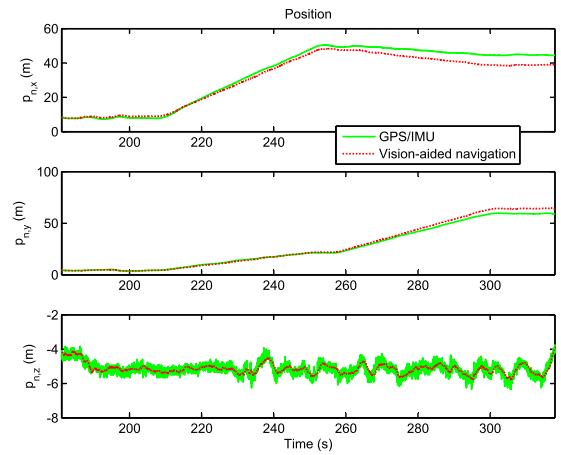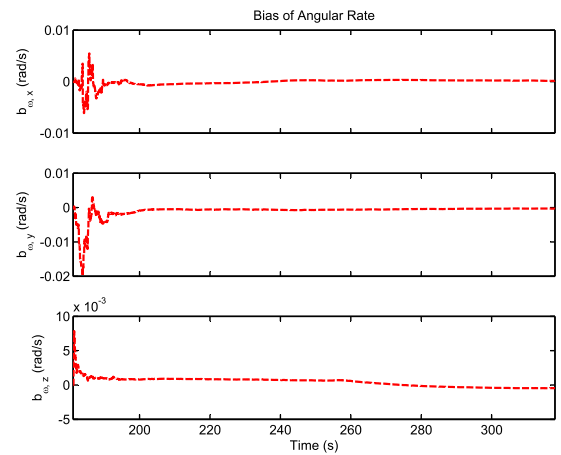

(b) Position
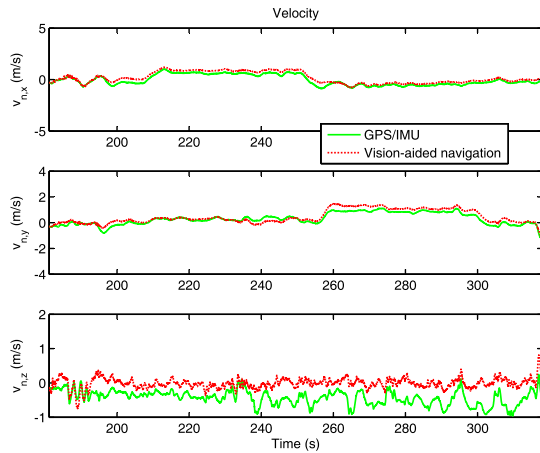
(c) Velocity

(d) Euler angle

(e) Bias of acceleration

(f) Bias of angular rate

Fig. 12 Closed-loop autonomous flight experimental results

6 Conclusions

This paper proposed a novel homography-based vision-aided inertial navigation system to provide drift-free velocity and attitude estimation for UAV stabilization control. The observability analysis of the proposed navigation system has shown that the velocity, attitude, and unknown biases are all observable when the UAV speed is nonzero. Comprehensive simulations and flight experiments have verified the effectiveness and robustness of the proposed navigation system. Since we assume that there is no map of the environment, the position estimation still drifts though the drift has been significantly reduced compared to pure inertial navigation. In practice, the proposed navigation system can be implemented independently. It can also be integrated with maps or aerial images to achieve drift-free position estimation.

References

1. Cai, G., Chen, B.M., Lee, T.H.: Unmanned Rotorcraft Systems. Springer, New York (2011)
2. Phang, S.K., Li, K., Yu, K.H., Chen, B.M., Lee, T.H.: Systematic design and implementation of a micro unmanned quadrotor system. Unmanned Syst. **2**(2), 121–141 (2014)
3. Kim, J., Sukkarieh, S.: Autonomous airborne navigation in unknown terrain environments. IEEE Trans. Aerosp. Electron. Syst. **40**, 1031–1045 (2004)
4. Wu, A.D., Johnson, E.N., Proctor, A.A.: Vision-aided inertial navigation for flight control. AIAA J. Aerosp. Comput. Inf. Commun. **2**(9), 348–360 (2005)
5. Kim, J., Sukkarieh, S.: Real-time implementation of airborne inertial-SLAM. Robot. Auton. Syst. **55**, 62–71 (2007)
6. Bryson, M., Sukkarieh, S.: Observability analysis and active control for airborne SLAM. IEEE Trans. Aerosp. Electron. Syst. **1**, 261–280 (2008)
7. Taylor, C.N., Veth, M.J., Raquet, J.F., Miller, M.M.: Comparison of two image and inertial sensor fusion techniques for navigation in unmapped environments. IEEE Trans. Aerosp. Electron. Syst. **47**(2), 946–958 (2011)
8. Zhang, J., Liu, W., Wu, Y.: Novel technique for vision-based UAV navigation. IEEE Trans. Aerosp. Electron. Syst. **47**(4), 2731–2741 (2011)
9. Scaramuzza, D., Fraundorfer, F.: Visual odometry: Part I - the first 30 years and fundamentals. IEEE Robot. Autom. Mag. **18**(4), 80–92 (2011)
10. Peng, K., Zhao, S., Lin, F., Chen, B.M.: Vision based target tracking/following and estimation of target motion. In: Proceedings of the 2013 AIAA Guidance, Navigation and Control Conference, Boston, USA (2013)
11. Lin, F., Ang, K.Z.Y., Wang, F., Chen, B.M., Lee, T.H., Yang, B., Dong, M., Dong, X., Cui, J., Phang, S.K., Wang, B., Luo, D., Peng, K., Cai, G., Zhao, S., Yin, M., Li, K.: Development of an unmanned coaxial rotorcraft for the darpa uavforge challenge. Unmanned Syst. **1**(2), 247–258 (2013)
12. Wang, F., Liu, P., Zhao, S., Chen, B.M., Phang, S.K., Lai, S., Pang, T., Wang, B., Cai, C., Lee, T.H.: Development of an unmanned helicopter for verticle replenishment. Unmanned Syst. **3**(1), 63–87 (2015)
13. Bonin-Font, F., Ortiz, A., Oliver, G.: Visual navigation for mobile robots: A survey. J. Intell. Robot. Syst. **53**, 263–296 (2008)
14. Conte, G., Doherty, P.: Vision-based unmanned aerial vehicle navigation using geo-referenced information. EURASIP J. Adv. Signal Process. 1–18 (2009)
15. Caballero, F., Merino, L., Ferruz, J., Ollero, A.: Vision-based odometry and SLAM for medium and high altitude flying UAVs. J. Intell. Robot. Syst. **54**, 137–161 (2009)
16. Kaiser, M.K., Gans, N.R., Dixon, W.E.: Vision-based estimation for guidance, navigation, and control of an aerial vehicle. IEEE Trans. Aerosp. Electron. Syst. **46**(3), 137–161 (2010)
17. Mourikis, A.I., Trawny, N., Roumeliotis, S.I., Johnson, A.E., Ansar, A., Matthies, L.: Vision-aided inertial navigation for spacecraft entry, descent, and landing. IEEE Trans. Robot. **25**(2), 264–280 (2009)
18. Howard, A.M., Jones, B.M., Serrano, N.: Integrated sensing for entry, descent, and landing of a robotic spacecraft. IEEE Trans. Aerosp. Electron. Syst. **47**(1), 295–304 (2011)
19. Wang, F., Cui, J., Phang, S.K., Chen, B.M., Lee, T.H.: A mono-camera and scanning laser range finder based UAV indoor navigation system. In: Proceedings of the 2013 International Conference on Unmanned Aircraft Systems, pp. 693–700. Atlanta, USA (2013)
20. Ma, Y., Soatto, S., Kosecka, J., Sastry, S.: An Invitation to 3D Vision. Springer, New York (2004)
21. Di, L., Fromm, T., Chen, Y.: A data fusion system for attitude estimation of low-cost miniature UAVs. J. Intell. Robot. Syst. (2011)
22. Shabayek, A.E.R., Demonceaux, C., Morel, O., Fofi, D.: Vision based UAV attitude estimation: Progress and insights. J. Intell. Robot. Syst. **65**(1–4), 295–308 (2012)
23. Martinelli, A.: Vision and IMU data fusion: Closed-form solutions for attitude, speed, absolute scale, and bias determination. IEEE Trans. Robot. **28**(1), 44–60 (2012)
24. Hartley, R.I., Zisserman, A.: Multiple View Geometry in Computer Vision, 2nd edn. Cambridge University Press, ISBN: 0521540518 (2004)
25. Groves, P.D.: Principles of GNSS, Inertial, and Multisensor Integrated Navigation Systems. Artech House (2008)
26. Nardone, S.C., Aidala, V.J.: Observability criteria for bearings-only target motion analysis. IEEE Trans. Aerosp. Electron. Syst. **17**(2), 162–166 (1981)
27. Nardone, S.C., Lindgren, A.G., Gong, K.F.: Fundamental properties and performance of conventional bearings-only target motion analysis. IEEE Trans. Autom. Control **29**(9), 775–787 (1984)
28. Zhao, S., Chen, B.M., Lee, T.H.: Optimal placement of bearing-only sensors for target localization. In: Proceedings

of the 2012 American Control Conference, pp. 5108–5113. Montreal, Canada (2012)

29. Zhao, S., Chen, B.M., Lee, T.H.: Optimal sensor placement for target localization and tracking in 2D and 3D. Int. J. Control. **86**(10), 1687–1704 (2013)

30. Michaelsen, E., Kirchhof, M., Stilla, U.: Sensor pose inference from airborne videos by decomposing homography estimates. In: Proceedings of the 2011 Asian Control Conference, pp. 211–216. Kaohsiung, Taiwan (2011)

31. Dong, X., Chen, B.M., Cai, G., Lin, H., Lee, T.H.: Development of a comprehensive software system for implementing cooperative control of multiple unmanned aerial vehicles. Int. J. Robot. Autom. **26**(1), 49–63 (2011)

32. Liu, P., Dong, X., Chen, B.M., Lee, T.H.: Development of an enhanced ground control system for unmanned aerial vehicles. In: Proceedings of the IASTED International Conference on Engineering and Applied Science, pp. 136–143. Colombo, Sri Lanka (2012)

Shiyu Zhao received the B.Eng. and M.Eng. degrees from Beijing University of Aeronautics and Astronautics, Beijing, China, in 2006 and 2009, respectively. He got his PhD degree in Electrical and Computer Engineering from National University of Singapore in 2014. His research interests lie in distributed control, estimation, and optimization in networked systems and its application to real-world intelligent and complex systems.

Feng Lin received the B.Eng. degree in computer science and control and the M.Eng. degree in system engineering from Beihang University, Beijing, China, in 2000 and 2003, respectively, and the Ph.D. degree in computer and electrical engineering from the National University of Singapore, Singapore, in 2011. He is a Senior Research Scientist with the Temasek Laboratories, National University of Singapore. His main research interests are unmanned aerial vehicles, vision-aided control and navigation, target tracking, robot vision, and embedded vision systems.

Kemao Peng received the B.Eng. degree in aircraft control systems, the M. Eng. degree in guidance, control, and simulation, and the Ph.D. degree in navigation, guidance and control, all from Beijing University of Aeronautics and Astronautics, Beijing, China, in 1986, 1989, and 1999, respectively. He is a Senior Research Scientist in the Temasek Laboratories, National University of Singapore. His research interests are robust control, nonlinear system control, autonomous flight control and vision based autonomous flight control.

Xiangxu Dong received the B.S. degree from Xiamen University, Xiamen, China, in 2006, and the PhD from National University of Singapore, Singapore, in 2013. He is now a Research Scientist with the Temasek Laboratories, National University of Singapore. His research interests include real-time software systems, formation flight control, and unmanned aerial vehicles.

Ben M. Chen received his B.S. in mathematics and computer science from Xiamen University, China, 1983, M.S. in Electrical Engineering from Gonzaga University, Spokane, Washington, 1988, and PhD in Electrical and Computer Engineering from Washington State University, Pullman, Washington, 1991. He is currently a Professor and Director of Control, Intelligent Systems & Robotics Area, Department of Electrical and Computer Engineering, National University of Singapore (NUS), and Head of Control Science Group, NUS Temasek Laboratories. His current research interests are in systems and control, unmanned aerial systems, and financial market modeling. Dr. Chen is an IEEE Fellow. He is the author/co-author of 10 research monographs. He had served on the editorial boards of a number of journals including IEEE Transactions on Automatic Control, Systems & Control Letters, and Automatica. He currently serves as an Editor-in-Chief of Unmanned Systems and a Deputy Editor-in-Chief of Control Theory & Technology.

Tong H. Lee received the B.A. degree (with first-class honors) from the University of Cambridge, Cambridge, U.K., in 1980, the M.Eng. degree from the National University of Singapore (NUS), Singapore, in 1985, and the Ph.D. degree from Yale University, New Haven, CT, USA, in 1987. He is a Professor in the Department of Electrical and Computer Engineering at the National University of Singapore (NUS); and also a Professor in the NUS Graduate School, NUS NGS. He was a Past Vice-President (Research) of NUS. Dr. Lee's research interests are in the areas of adaptive systems, knowledge-based control, intelligent mechatronics and computational intelligence. He currently holds Associate Editor appointments in the IEEE Transactions in Systems, Man and Cybernetics; IEEE Transactions in Industrial Electronics; Control Engineering Practice (an IFAC journal); and the International Journal of Systems Science (Taylor and Francis, London). In addition, he is the Deputy Editor-in-Chief of IFAC Mechatronics journal.