Open access • Journal Article • DOI:10.1007/S10514-017-9690-5

Autonomous navigation of micro aerial vehicles using high-rate and low-cost sensors — Source link

Angel Santamaria-Navarro, Giuseppe Loianno, Joan Sola, Vijay Kumar ...+1 more authors

Institutions: Spanish National Research Council, University of Pennsylvania

Published on: 01 Jun 2018 - Autonomous Robots (Springer US)

Topics: Odometry, Smart camera, Inertial measurement unit, Optical flow and Kalman filter

Noname manuscript No.
(will be inserted by the editor)

Autonomous navigation of micro aerial vehicles using high-rate and low-cost sensors

Angel Santamaria-Navarro · Giuseppe Loianno · Joan Solà · Vijay Kumar · Juan Andrade-Cetto

Received: 21 July 2016 / Accepted: 20 November 2017

Abstract The combination of visual and inertial sensors for state estimation has recently found wide echo in the robotics community, especially in the aerial robotics field, due to the lightweight and complementary characteristics of the sensors data. However, most state estimation systems based on visual-inertial sensing suffer from severe processor requirements, which in many cases make them impractical. In this paper, we propose a simple, low-cost and high rate method for state estimation enabling autonomous flight of Micro Aerial Vehicles (MAVs), which presents a low computational burden. The proposed state estimator fuses observations from an inertial measurement unit, an optical flow smart camera and a time-of-flight range sensor. The smart camera provides optical flow measurements up to a rate of 200 Hz, avoiding the computational bottleneck to the main processor produced by all image processing requirements. To the best of our knowledge, this is the first example of extending the use of these smart cameras from hovering-like motions to odometry estimation, producing estimates that are usable during flight times of several minutes. In order to validate and defend the simplest algorithmic solution, we investigate the performances of two Kalman filters, in the extended and error-state flavors, alongside with a large number of algorithm modifications defended in earlier literature on visual-inertial odometry, showing that their impact on filter performance is minimal. To close the control loop, a non-linear controller operating in the special Euclidean group $SE(3)$ is able to drive, based on the estimated vehicle's state, a quadrotor platform in 3D space guaranteeing the asymptotic stability of 3D position and heading. All the estimation and control tasks are solved on board and in real time on a limited computational unit.

The proposed approach is validated through simulations and experimental results, which include comparisons with ground-truth data provided by a motion capture system. For the benefit of the community, we make the source code public.

Keywords Micro Aerial Vehicles · Vision for Robotics · Localization

This work was funded by the project ROBINSTRUCT (TIN2014-58178-R) and the Ramón y Cajal posdoctoral fellowship (RYC-2012-11604) from the Spanish Ministry of Economy and Competitiveness, by the Spanish State Research Agency through the María de Maeztu Seal of Excellence to IRI (MDM-2016-0656), by the EU H2020 project AEROARMS (H2020-ICT-2014-1-644271) and by the ARL grant W911NF-08-2-0004, the ONR grants N00014-07-1-0829, N00014-14-1-0510, the ARO grant W911NF-13-1-0350, the NSF grants IIS-1426840, IIS-1138847, and by the DARPA grants HR001151626, and HR0011516850.

The paper has supplementary downloadable material available at http://www.springer.com, provided by the authors. The material includes a video showing experimental results of the methods presented in the paper.

A. Santamaria-Navarro, J. Solà and J. Andrade-Cetto
Institut de Robòtica i Informàtica Industrial, CSIC-UPC
Llorens Artigas 4-6, Barcelona 08028, Spain
Tel.: +34-93-4015751, Fax: +34-93-4015750
E-mail: {asantamaria, jsola, cetto}@iri.upc.edu

Giuseppe Loianno and Vijay Kumar
GRASP Lab, University of Pennsylvania
3330 Walnut Street, 19104 Philadelphia, USA
Tel.: +1-215-898-3630, Fax: +1-215-573-6334
E-mail: {loiannog, kumar}@seas.upenn.edu

1 Introduction

Small size unmanned aerial vehicles, namely Micro Aerial Vehicles (MAVs), have gained significant attention in the last decade both in academia and industry,

mostly due to their potential use in a wide range of applications such as exploration (Tomic et al., 2012), inspection (Ozaslan et al., 2013), mapping, interaction with the environment (Forte et al., 2012; Thomas et al., 2014), search and rescue (Michael et al., 2012), and to their significant mechanical simplicity (Michael et al., 2010) and ease of control. Moreover, their ability to operate in confined spaces, hover in space and even perch, together with a decrease in cost make them very attractive with tremendous potential. To enable autonomous missions with MAVs, a robust, accurate and high update rate state estimation pipeline is crucial.

Low-cost, low complexity solutions for MAV state estimation are not very common. A first family of localization systems relies on external infrastructure (GPS, RF beacons, visual tags, IR cameras), *e.g.* (Liu et al., 2007), which may not always be practical. When using low-cost setups, these systems drastically lack precision, dynamics, or both. A second family of methods embark all hardware and software for self-localization, thus not relying on any external setup. These methods include any kind of Simultaneous Localization And Mapping (SLAM) or odometry systems drawn from other robotics fields to be applied to MAVs. Good results have been obtained using stereo camera configurations (Shen et al., 2013; Tomic et al., 2012), and RGB-D sensor systems (Shen et al., 2012; Loianno et al., 2015b; Michael et al., 2012). However, these algorithms generally require powerful computers to produce and deal with fairly dense point clouds. State estimation for small-sized and lightweight platforms (less than 40 cm and less than 1 kg) is a challenging task, due to the limited payload and computing capacity that these vehicles can carry.

Not surprisingly, combinations of Inertial Measurement Units (IMU) and monocular visual sensors are becoming very popular, thanks to their low weight, power consumption and cost, and their ease of installation. This constitutes a minimalist yet powerful sensor suite for autonomous localization as it allows recovering both the high motion dynamics and the localization with respect to the environment, including scale and, most important for MAV navigation, the direction of gravity (Jones and Soatto, 2011; Meier et al., 2012; Martinelli, 2013). The processes of estimating the vehicle state using such sensors are known as Visual-Inertial Odometry (VIO, with no absolute localization) (Roumeliotis et al., 2002; Hesch et al., 2014; Li and Mourikis, 2013), and Visual-Inertial SLAM (enabling absolute localization by revisiting previously mapped areas) (Kelly and Sukhatme, 2011; Blösch et al., 2010; Roussillon et al., 2011; Fraundorfer et al., 2012). The focus of our work is not at building maps, and we concentrate on VIO.

In most of the above-mentioned VIO approaches, the state vector being estimated is very large. Their typical high precision outcome is attained by jointly estimating a subset of past camera poses and a number of landmarks in the environment, which are tracked in the image over relatively long periods of time. This creates a graph of constraints between vehicle states and landmarks, which is incrementally solved by nonlinear optimization. These solutions are computationally intensive, and require carefully optimized code to compensate for the limited computing resources typical of MAVs.

To reduce the computational burden and increase the update rate, several authors opt to exploit the image information only locally and in 2D. Forster et al. (2014) define a method called semi-direct visual odometry (SVO) which establishes feature correspondences directly from intensity values in the image (small patches) rather than feature extraction and matching. This icreases speed due to the lack of feature extraction at every frame and accuracy through subpixel feature correspondence, allowing its implementation onboard an aerial platform (Faessler et al., 2016). However, in Forster et al. (2014) feature extraction is still required when a keyframe is selected to initialize new 3D points. Weiss et al. (2012) propose an EKF-based speed estimation module, which converts the pair camera-IMU into a metric speed sensor at 40 Hz update rate, using optical flow information of at least 2 image features. A parallel tracking and mapping (PTAM) pipeline is tailored to the system, which therefore requires onboard image processing. In Omari and Ducard (2013), flow information is fused with inertial measurements. However, only simulation results are provided. A similar approach (Blösch et al., 2014) presents a novel visual error term and uses a visual-inertial sensor consisting on a synchronized global-shutter camera and IMU (Nikolic et al., 2014) to obtain flow information, though running at 20Hz.

In this paper, we present a fast and low-cost state estimation method for small-sized MAVs. We use exclusively low-cost sensors and low-complexity algorithms. As hardware, we take advantage of a low-cost IMU (ASCTEC Hummingbird built-in IMU), a smart camera (Honegger et al., 2013) which directly outputs 2D flow, and an infrared time-of-flight range sensor (Ruffo et al., 2014), all shown in Fig. 1. This sensor setup has the advantage of being simple, lightweight, low power and low cost, and is already included, with minor variations, in several commercial multi-copters as their basic instrumentation, being typically used as a means

Fig. 1 Bottom view of the ASCTEC Hummingbird quadrotor used in the experiments. The vehicle is equipped with a built-in IMU, a smart camera and a time-of-flight range sensor.

for automatic hovering. As software, we have developed two Kalman filters, in the extended (EKF) and error-state (ESKF) forms (Ravindra et al., 2012), together with a wide range of variations in the inner details, for the sake of performance evaluation and comparison. The overall estimation system acts as an odometer that provides absolute altitude, velocity, orientation, angular rate, and acceleration, with respect to a precise gravity reference, at a rate of 100 Hz. The x and y positions and the *yaw* angle are not observable, and their output is the result of an incremental estimation subject to drift —these modes can be observed with a lower update rate by a higher level task, such as a visual servoing (Santamaria-Navarro et al., 2014, 2017; Rossi et al., 2017).

When compared to other visual-inertial approaches, the optical-flow nature of the smart camera, with a very narrow field of view (FoV) of only 64×64 pixels or 1.6° (compared to the 90° typical of VIO), represents an important limitation, in the sense that visual features are only exploited locally both in space and time, *i.e.*, there is no notion of multiple features being tracked over long periods of time. The number and length of the feature tracks are key to the high precision attained by the more sophisticated VIO methods, and in consequence, we cannot expect equivalent performances. By contrast, the number and length of these feature tracks are the ones responsible for the algorithmic complexity and CPU consumption. In our case, the high filter update rate, made possible by the smart camera and range sensor, and by our light algorithm, contributes to decrease the sensitivity to linearization errors, reducing the accumulation of drift and thus enabling much simpler methods for an acceptable performance.

In this scenario, one key objective of this paper is to show that, given a sensor setup with such capabilities, we are able to derive motion estimates that are useful in the mid term (a few minutes, *i.e.*, the typical flight times of a full battery recharge) to drive autonomously the vehicle, without the need to implement complex algorithms. In order to defend the simplest estimation solution, we benchmark a large number of algorithm improvements and variations described in the literature, and show that their impact on system performance is minimal. This should not be surprising given the high frequency of the measurements, but we believe that our benchmarking provides valuable results for establishing good estimation practices.

Filter type	Quat. error	Quat. integration	Trans. Mat. Trunc.
ESKF	GE, LE	Q0F, Q0B, Q1	$\mathbf{F}_1, \mathbf{F}_2, \mathbf{F}_3$
EKF	–	Q0F, Q0B, Q1	$\mathbf{F}_1, \mathbf{F}_2, \mathbf{F}_3$

Table 1 Kalman filters and algorithm variations

The algorithm variations that we investigate are shown in Table 1, and are properly defined later in the text. They are summarized hereafter, together with the key works that defended them. First, we implement error-state (ESKF) and extended (EKF) Kalman filters (Madyastha et al., 2011). Second, we express the orientation errors of ESKF both in local (LE) and global (GE) frames (Li and Mourikis, 2012).[1] Third, we compare different schemes for rotational motion integration of the quaternion (Trawny and Roumeliotis, 2005), including forward zeroth-order (Q0F), backward zeroth-order (Q0B), and first-order (Q1). Fourth, we compare different integration forms and approximations of the system's transition matrix ($\mathbf{F}_1, \mathbf{F}_2, \mathbf{F}_3$) (Li and Mourikis, 2013).

In order to achieve fully autonomous capabilities, we use the estimated state to feed a closed-loop flight controller at 100Hz. Many control approaches have been proposed to drive MAVs in 3D space. In most previous works, a backstepping approach is used for control because the attitude dynamics can be assumed to be faster than the dynamics governing the position. Thus, linearized controllers are used for both loops (Michael et al., 2010; Weiss et al., 2011; Hérissé et al., 2012). In this work, we use a nonlinear controller that operates on the special Euclidean group $SE(3)$, based on the work (Lee et al., 2013; Mellinger and Kumar, 2011). This allows us to accurately control large excursions from the hover position during operations like take-off and navigation, enabling agile flight maneuvers with robustness.

In this paper we provide the following contributions with respect to our previous conference paper (Santamaria-Navarro et al., 2015), in which we originally addressed the VIO problem. First, we define the observation function directly in the flow space instead

[1] Note that in EKF the orientation error is additive and this distinction is irrelevant.

of using pre-processed planar velocities. Second, we substitute the sonar-based ranger with an IR time-of-flight ranger. This required the development of a new observation model. And third, we add $SE(3)$-based closed-loop control using the state estimates to track 3D trajectories planned to be dynamically feasible.

The entire estimation and control solutions are able to run using half of the computational power of a limited computational unit, the Odroid-XU3[2]. To the best of our knowledge, these are the first examples of the usage of such a low-cost visual-range-inertial sensor setup for 6 Degree of Freedom (DoF) motion estimation and control.

This paper is organized as follows. The next section develops our state estimation approach by describing the sensory setup, system kinematics and main filtering details. In Section 3 we describe the non-linear control law and the dynamically feasible trajectory planner. Section 4 presents the simulations and experimental results that validate the proposed methods. Finally, Section 5 concludes the work and provides an overview of the multiple future scenarios. Some accessory or complementary mathematical derivations supporting the analysis are provided in the Appendices.

2 Flow-Inertial-Range Odometry

We consider a quadrotor, equipped with IMU, flow smart camera and infrared (IR) range sensors, moving with respect to a global coordinate frame assumed inertial. All sensors are rigidly attached together below the aerial platform, with the smart camera and range sensor pointing downwards. Their models and characteristics are specified in the following sections. The platform or body frame is defined as the IMU frame, and the other sensor frames are calibrated offline with respect to it. All the image processing to obtain the flow is done by the smart camera.

We aim at tracking the vehicle's state by integrating IMU measurements, and to correct these estimates with flow and range readings, observing in turn the IMU biases for their proper compensation. We then use the estimated state to perform closed-loop control of the aerial platform. An overview of the architecture is depicted in Fig. 2.

[2] http://www.hardkernel.com/main/products/prdt_info.php?g_code=G140448267127

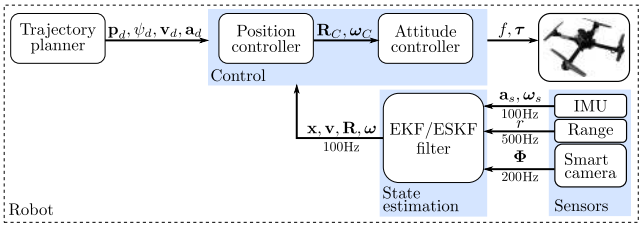


Fig. 2 Overview of the architecture pipeline for estimation and control

2.1 Filter modalities and state vectors

In order to describe the state estimation formulation for ESKF and EKF in subsequent sections, we present here the following definitions.

In ESKF formulations, we speak of true-, nominal- and error-state values, where the error-state values represent the discrepancy between the nominal- and the true-state values. We note the true states with a 't' subindex, $\mathbf{x}_t$; nominals with the plain symbol, $\mathbf{x}$; and errors with a 'δ' prefix, $\delta\mathbf{x}$. These are defined respectively as,

$$\mathbf{x}_t = \begin{bmatrix}\mathbf{p}_t\ \mathbf{v}_t\ \mathbf{q}_t\ \mathbf{a}_{bt}\ \boldsymbol{\omega}_{bt}\end{bmatrix}^\top \in \mathbb{R}^{16} \tag{1a}$$

$$\mathbf{x} = \begin{bmatrix}\mathbf{p}\ \mathbf{v}\ \mathbf{q}\ \mathbf{a}_b\ \boldsymbol{\omega}_b\end{bmatrix}^\top \in \mathbb{R}^{16} \tag{1b}$$

$$\delta\mathbf{x} = \begin{bmatrix}\delta\mathbf{p}\ \delta\mathbf{v}\ \delta\boldsymbol{\theta}\ \delta\mathbf{a}_b\ \delta\boldsymbol{\omega}_b\end{bmatrix}^\top \in \mathbb{R}^{15}\,, \tag{1c}$$

where $\mathbf{p}$, $\mathbf{v}$, $\mathbf{q}$ are position, velocity and quaternion orientation (refer to App. A for quaternion conventions), all expressed in global world (inertial) frame, and $\mathbf{a}_b$ and $\boldsymbol{\omega}_b$ are accelerometer and gyrometer biases respectively. The error-state is modeled as a Gaussian distribution $\delta\mathbf{x} \sim \mathcal{N}\{\widehat{\delta\mathbf{x}}, \mathbf{P}\}$. These states are related by the composition

$$\mathbf{x}_t = \mathbf{x} \oplus \delta\mathbf{x}\,, \tag{2}$$

where $\oplus$ wraps the error $\delta\mathbf{x}$ onto the manifold of $\mathbf{x}$. It corresponds to the trivial addition for all state variables (*e.g.* $\mathbf{p}_t = \mathbf{p} + \delta\mathbf{p}$) except for the orientation. We use a minimal orientation error $\delta\boldsymbol{\theta} \in \mathfrak{so}3 \subset \mathbb{R}^3$ living in the space tangent to the $SO(3)$ manifold (*i.e.*, in its Lie algebra $\mathfrak{so}(3)$). We contemplate orientation error definitions in the global frame (GE) or in the local frame (LE); their composition is computed respectively with a product *on the left or right hand sides* of the nominal quaternion,

$$\text{global error (GE):}\quad \mathbf{q}_t = \delta\mathbf{q} \otimes \mathbf{q} \tag{3a}$$

$$\text{local error (LE):}\quad \mathbf{q}_t = \mathbf{q} \otimes \delta\mathbf{q}\,, \tag{3b}$$

where $\delta\mathbf{q} = \mathbf{q}\{\delta\boldsymbol{\theta}\} \triangleq \exp(\delta\boldsymbol{\theta}/2)$ is the orientation error in $SO(3)$ expressed as a unit quaternion —see App. A for details.

In EKF formulations, we directly estimate the true-state, which is modeled as a Gaussian distribution $\mathbf{x}_t \sim \mathcal{N}\{\widehat{\mathbf{x}}_t, \mathbf{P}\}$.

2.2 Observation models

The observation models described hereafter are used in the filter propagation (IMU) and correction (smart camera and range sensor) steps.

2.2.1 Inertial measurement unit (IMU)

The IMU is composed of a 3-axis accelerometer and a 3-axis gyrometer providing, in body frame, acceleration and angular velocity measurements respectively. The accelerometer measures acceleration $\mathbf{a}_t$ and gravity $\mathbf{g}$ (considered known and constant) together. The gyrometer measures angular rates $\boldsymbol{\omega}_t$, which are already in body frame. These measurements are affected by additive biases, $\mathbf{a}_{bt}$ and $\boldsymbol{\omega}_{bt}$, and noises $\mathbf{a}_n$ and $\boldsymbol{\omega}_n$. The IMU model reads,

$$\mathbf{a}_S = \mathbf{R}_t^\top (\mathbf{a}_t - \mathbf{g}) + \mathbf{a}_{bt} + \mathbf{a}_n \quad \in \mathbb{R}^3 \tag{4a}$$

$$\boldsymbol{\omega}_S = \boldsymbol{\omega}_t + \boldsymbol{\omega}_{bt} + \boldsymbol{\omega}_n \quad \in \mathbb{R}^3\,, \tag{4b}$$

where $\mathbf{R}_t \triangleq \mathbf{R}\{\mathbf{q_t}\}$ is the rotation matrix equivalent to the quaternion $\mathbf{q_t}$. The noises are modeled by Gaussian distributions $\mathbf{a}_n \sim \mathcal{N}\{\mathbf{0}, \mathbf{A}_n\}$ and $\boldsymbol{\omega}_n \sim \mathcal{N}\{\mathbf{0}, \boldsymbol{\Omega}_n\}$.

2.2.2 Optical flow smart camera

The smart camera (PX4-Flow) integrates a monocular camera, an ultrasound range sensor aligned with the optical axis, and a triaxial gyrometer. It provides raw optical flow (from now on referred to as just *flow*) at the principal point, a range to a reflective surface, and 3-axial angular rates. In addition, it delivers horizontal velocities in metric scale, computed from all the input data. In this work, we improve the solution presented in our previous work (Santamaria-Navarro et al., 2015) with the following modifications. We discard the internally computed linear velocities and take advantage of the raw flow to define a better observation function directly in the flow space. In contrast to these linear velocities, the raw flow is not dependent on the filter state thus the process and measurement noises are decoupled. We also discard the PX4's angular rates and range. For the angular rates, we use the quadrotor's IMU gyrometers because they are of higher quality and already aligned with the vehicle frame. As for the range, we replace the sonar by an IR time-of-flight ranger.

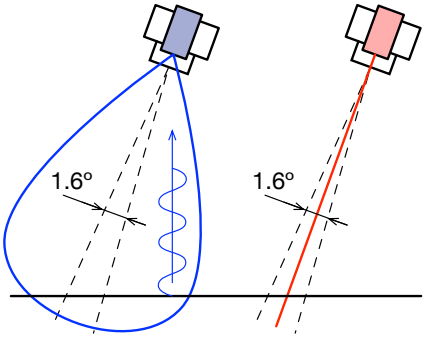


Fig. 3 Field of view (FoV) and echo directions using sonar (left) and IR (right) rangers, compared to the camera FoV of 1.6° (dashed). Due to the coarse directivity of the sonar, the strongest echoes are likely to come from a direction perpendicular to the terrain surface, which may fall out of the FoV of the camera, creating false visual-range associations on tilt maneuvers. This is avoided with the excellent directivity of the IR ranger.

Due to the coarse directivity of the sonar, the strongest echoes are likely to come from a direction perpendicular to the terrain surface, which has two main implications during tilt maneuvers. First, it can produce numerous outliers (see Figure 4 of (Santamaria-Navarro et al., 2015)). Second, the signal bouncing may fall out of the FoV of the camera, creating false visual-range associations. These are avoided with the excellent directivity of the IR ranger, which has less outliers and ensures that the range echo corresponds to the region observed by the camera (1.6° FoV), resulting in a more predictable behavior that better fits the observation model. In order for the smart camera to perform correctly, scene lighting must be adequate and the ground is required to have good visual texture.

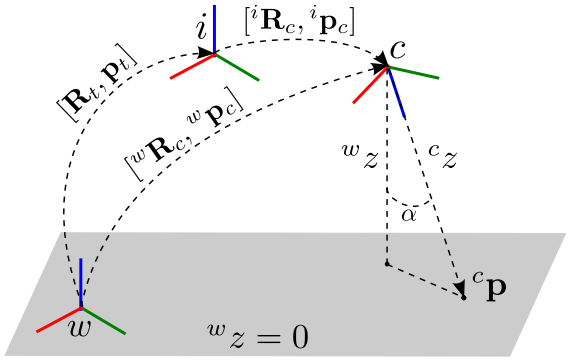


Fig. 4 World (w), IMU (i) and camera (c) frames and their relative transformations (curved arrows). The IMU and camera are attached together forming a rigid body, with a twist $\{\mathbf{v}_i, \boldsymbol{\omega}_i\}$ at the IMU, which induces a twist $\{\mathbf{v}_c, \boldsymbol{\omega}_c\}$ at the camera. A point $\mathbf{p}$ on the ground is defined along the camera's optical axis (its Z axis).

The observation model of the flow and the required background are described in the following. Let w, i and c denote respectively the world, IMU and camera frames. Let ${}^c\mathbf{p} = \begin{bmatrix} {}^cx & {}^cy & {}^cz \end{bmatrix}^\top$ be a static 3D point in the ground (${}^wz = 0$), expressed in the camera reference frame c as shown in Fig. 4. This point is projected to the image according to the pin-hole model,

$$\boldsymbol{\pi} = \mathbf{P}_f \, \frac{{}^c\mathbf{p}}{{}^cz} \,, \tag{5}$$

with $\mathbf{P}_f$ the projection matrix defined in (10) and ${}^cz$ the distance, measured along the optical axis, from the optical center to the ground . Taking its time derivative we obtain the optical flow,

$$\boldsymbol{\phi} \triangleq \dot{\boldsymbol{\pi}} = \mathbf{P}_f \, \frac{{}^c\dot{\mathbf{p}}\,{}^cz - {}^c\mathbf{p}\,{}^c\dot{z}}{{}^cz^2} \,. \tag{6}$$

The smart camera computes the mean flow in a 64×64 pixels patch centered at the principal point, giving a FoV of 1.6° around the optical axis. At the optical axis, we have ${}^c\mathbf{p} = [0, 0, {}^cz]$ and (6) reduces to

$$\boldsymbol{\phi} = \mathbf{P}_f \, \frac{{}^c\dot{\mathbf{p}}}{{}^cz} \,. \tag{7}$$

Considering the smart camera in motion with a twist ${}^c\mathbf{v}_c, {}^c\boldsymbol{\omega}_c$ in its own frame, the velocity of the point in the camera frame is

$${}^c\dot{\mathbf{p}} = -{}^c\mathbf{v}_c - {}^c\boldsymbol{\omega}_c \times {}^c\mathbf{p} \,. \tag{8}$$

Injecting this in (7) leads after easy rearrangements to

$$\boldsymbol{\phi} = -\mathbf{P}_f \, \frac{{}^c\mathbf{v}_c}{{}^cz} + \mathbf{P}_\times \, {}^c\boldsymbol{\omega}_c \,, \tag{9}$$

where the matrices $\mathbf{P}_f$ and $\mathbf{P}_\times$ can be expressed in terms of the camera's focal distances (f_x, f_y), measured in horizontal and vertical pixels respectively, with

$$\mathbf{P}_f = \begin{bmatrix} f_x & 0 & 0 \\ 0 & f_y & 0 \end{bmatrix} \,, \qquad \mathbf{P}_\times = \begin{bmatrix} 0 & f_x & 0 \\ -f_y & 0 & 0 \end{bmatrix} \,. \tag{10}$$

To obtain the observation model depending on our system variables, we can expand ${}^c\mathbf{v}_c$, ${}^c\boldsymbol{\omega}_c$ and ${}^cz$ with

$${}^c\mathbf{v}_c = {}^i\mathbf{R}_c^\top \left(\mathbf{R}_t^\top \, \mathbf{v}_t + \boldsymbol{\omega}_t \times {}^i\mathbf{p}_c \right) \tag{11a}$$

$${}^c\boldsymbol{\omega}_c = {}^i\mathbf{R}_c^\top \, \boldsymbol{\omega}_t \tag{11b}$$

$${}^cz = \frac{{}^wz_c}{\cos\alpha} = \frac{{}^w\mathbf{p}_c(3)}{{}^w\mathbf{R}_c(3,3)} \,, \tag{11c}$$

where the true angular velocity $\boldsymbol{\omega}_t$ is obtained using (4b), the pair $\langle {}^i\mathbf{p}_c, {}^i\mathbf{R}_c \rangle$ is the calibrated camera pose expressed in IMU frame, α is the angle between the Z axes of the world and camera frames (Fig. 4), and

$${}^w\mathbf{p}_c = \mathbf{p}_t + \mathbf{R}_t \, {}^i\mathbf{p}_c \tag{12a}$$

$${}^w\mathbf{R}_c = \mathbf{R}_t \, {}^i\mathbf{R}_c \,. \tag{12b}$$

The final observation model considers noisy flow measurements with a noise $\mathbf{n}_\phi \sim \mathcal{N}\{\mathbf{0}, \mathbf{N}_\phi\}$,

$$h_\phi(\mathbf{x}_t, \mathbf{n}_\phi) = \boldsymbol{\phi}(\mathbf{x}_t) + \mathbf{n}_\phi \quad \in \mathbb{R}^2 \,. \tag{13}$$

2.2.3 Range sensor

The infrared time-of-flight sensor provides the distance to the surface that bounces its signal (Fig. 3-right). When mounted under a quadrotor, facing down and assuming there is no object below the platform, the sensor model is similar to the range presented in (11c), but using the ranger frame instead of the camera frame,

$$h_r(\mathbf{x}_t, n_r) = \frac{{}^w\mathbf{p}_r(3)}{{}^w\mathbf{R}_r(3,3)} + n_r \quad \in \mathbb{R} \,, \tag{14}$$

with

$${}^w\mathbf{p}_r = \mathbf{p}_t + \mathbf{R}_t \, {}^i\mathbf{p}_r \tag{15a}$$

$${}^w\mathbf{R}_r = \mathbf{R}_t \, {}^i\mathbf{R}_r \,, \tag{15b}$$

with $n_r \sim \mathcal{N}\{0, N_r\}$ and where the pair $\langle {}^i\mathbf{p}_r, {}^i\mathbf{R}_r \rangle$ is the calibrated pose of the range sensor with respect to the IMU frame. Notice the model difference with respect to a sonar (Santamaria-Navarro et al., 2015) which directly retrieves the platform height.

2.3 System kinematics

As it is common practice in the literature of IMU navigation, *e.g.* (Roussillon et al., 2011), we can define the continuous system kinematic equations $\dot{\mathbf{x}}_t = f(\mathbf{x}_t, \mathbf{u}, \mathbf{w})$ as

$$\dot{\mathbf{p}}_t = \mathbf{v}_t \tag{16a}$$

$$\dot{\mathbf{v}}_t = \mathbf{R}_t \left(\mathbf{a}_S - \mathbf{a}_{bt} - \mathbf{a}_n \right) + \mathbf{g} \,, \tag{16b}$$

$$\dot{\mathbf{q}}_t = \frac{1}{2} \, \mathbf{q}_t \otimes \left(\boldsymbol{\omega}_S - \boldsymbol{\omega}_{bt} - \boldsymbol{\omega}_n \right) \tag{16c}$$

$$\dot{\mathbf{a}}_{bt} = \mathbf{a}_w \tag{16d}$$

$$\dot{\boldsymbol{\omega}}_{bt} = \boldsymbol{\omega}_w \,. \tag{16e}$$

where we use the shortcut $\mathbf{q} \otimes \boldsymbol{\omega} \equiv \mathbf{q} \otimes [0, \boldsymbol{\omega}]^\top$ for convenience of notation. This system involves the true-state $\mathbf{x}_t$ from (1a), is governed by IMU noisy readings $\mathbf{u} = \begin{bmatrix} \mathbf{a}_S & \boldsymbol{\omega}_S \end{bmatrix}^\top$ (4), and is perturbed by Gaussian noise $\mathbf{w} = \begin{bmatrix} \mathbf{a}_w & \boldsymbol{\omega}_w \end{bmatrix}^\top$, with $\mathbf{a}_w \sim \mathcal{N}\{\mathbf{0}, \mathbf{A}_w\}$ and $\boldsymbol{\omega}_w \sim \mathcal{N}\{\mathbf{0}, \boldsymbol{\Omega}_w\}$.

In ESKF only, we distinguish between nominal- and error-state kinematics. The nominal kinematics correspond to the modeled system without noises or perturbations

$$\dot{\mathbf{p}} = \mathbf{v} \tag{17a}$$

$$\dot{\mathbf{v}} = \mathbf{R} \left(\mathbf{a}_S - \mathbf{a}_b \right) + \mathbf{g} \,, \tag{17b}$$

$$\dot{\mathbf{q}} = \frac{1}{2} \, \mathbf{q} \otimes \left(\boldsymbol{\omega}_S - \boldsymbol{\omega}_b \right) \tag{17c}$$

$$\dot{\mathbf{a}}_b = \mathbf{0} \tag{17d}$$

$$\dot{\boldsymbol{\omega}}_b = \mathbf{0} \,. \tag{17e}$$

To obtain the error kinematics we write each true-state equation in (16) as its composition of nominal- and error-states in (2) and (3), solve for the error-state, and simplify all second-order infinitesimals. The result for the linear velocity and orientation elements ($\delta\mathbf{v}$ and $\delta\boldsymbol{\theta}$) depend on the orientation error representation (GE or LE). With a globally-defined error GE we have

$$\delta\dot{\mathbf{p}} = \delta\mathbf{v} \tag{18a}$$
$$\delta\dot{\mathbf{v}} = -[\mathbf{R}(\mathbf{a}_S - \mathbf{a}_b)]_\times \delta\boldsymbol{\theta} - \mathbf{R}\,\delta\mathbf{a}_b - \mathbf{R}\mathbf{a}_n \tag{18b}$$
$$\delta\dot{\boldsymbol{\theta}} = -\mathbf{R}\delta\boldsymbol{\omega}_b - \mathbf{R}\,\boldsymbol{\omega}_n \tag{18c}$$
$$\delta\dot{\mathbf{a}}_b = \mathbf{a}_w \tag{18d}$$
$$\delta\dot{\boldsymbol{\omega}}_b = \boldsymbol{\omega}_w\,, \tag{18e}$$

whereas with a local definition LE we need to replace (18b) and (18c) above with

$$\delta\dot{\mathbf{v}} = -\mathbf{R}\,[\mathbf{a}_S - \mathbf{a}_b]_\times \delta\boldsymbol{\theta} - \mathbf{R}\,\delta\mathbf{a}_b - \mathbf{R}\mathbf{a}_n \tag{18f}$$
$$\delta\dot{\boldsymbol{\theta}} = -[\boldsymbol{\omega}_S - \boldsymbol{\omega}_b]_\times \delta\boldsymbol{\theta} - \delta\boldsymbol{\omega}_b - \boldsymbol{\omega}_n\,, \tag{18g}$$

where $[\cdot]_\times$ is the skew-symmetric matrix

$$[\mathbf{a}]_\times = \begin{bmatrix} 0 & -a_z & a_y \\ a_z & 0 & -a_x \\ -a_y & a_z & 0 \end{bmatrix}\,. \tag{19}$$

2.3.1 System kinematics in discrete time

One of the aims of this paper is to analyze the impact of using different integration approximations of the previous equations. Integrating continuous differential equations of the type $\dot{\mathbf{x}} = f(\mathbf{x}, \mathbf{u})$ from time $(k-1)\Delta t$ to $k\Delta t$ can be done in a number of ways. A common technique is to integrate the linearized system, $\dot{\mathbf{x}} = \mathbf{A}\,\mathbf{x} + \mathbf{B}\,\mathbf{u}$, with $\mathbf{A} = \partial f/\partial\mathbf{x}$, $\mathbf{B} = \partial f/\partial\mathbf{u}$, into the discrete-time $\mathbf{x}_k \approx \mathbf{F}\,\mathbf{x}_{k-1} + \mathbf{B}\,\mathbf{u}\Delta t$, with $\mathbf{F} = e^{\mathbf{A}\Delta t}$, and to truncate the exponential Taylor series $e^{\mathbf{A}\Delta t} = \sum \mathbf{A}^n \Delta t^n/n!$ at different orders, obtaining the different approximations $\mathbf{F}_N$ of the filter transition matrix,

$$\begin{aligned}\mathbf{F}_N &\triangleq \sum_{n=0}^{N<\infty} \frac{1}{n!}\mathbf{A}^n\Delta t^n \\ &= \mathbf{I} + \mathbf{A}\Delta t + \frac{1}{2}\mathbf{A}^2\Delta t^2 + ... + \frac{1}{N!}\mathbf{A}^N\Delta t^N\,,\end{aligned} \tag{20}$$

with $\mathbf{I}$ the identity matrix. We provide extensive details in App. B.

For the quaternion, it is possible and convenient, through the exponential maps, to obtain closed-form expressions of the infinite Taylor series (Trawny and Roumeliotis, 2005; Solà, 2015). For this reason, we contemplate here the *zero-th forward*, *zero-th backward* and *first* order integrators of (16c), that we name Q0F, Q0B and Q1 respectively,

$$\text{Q0F: } \mathbf{q}_k \approx \mathbf{q}_{k-1} \otimes \mathbf{q}\{\boldsymbol{\omega}_{k-1}\Delta t\} \tag{21a}$$
$$\text{Q0B: } \mathbf{q}_k \approx \mathbf{q}_{k-1} \otimes \mathbf{q}\{\boldsymbol{\omega}_k\Delta t\} \tag{21b}$$
$$\text{Q1: } \mathbf{q}_k \approx \mathbf{q}_{k-1} \otimes \left(\mathbf{q}\{\bar{\boldsymbol{\omega}}\Delta t\} + \frac{\Delta t^2}{24}\begin{bmatrix} 0 \\ \boldsymbol{\omega}_{k-1}\times\boldsymbol{\omega}_k \end{bmatrix}\right), \tag{21c}$$

with $\boldsymbol{\omega}_k \triangleq \boldsymbol{\omega}_{S,k} - \boldsymbol{\omega}_{bt,k}$, $\boldsymbol{\omega}_{bt,k} \approx \boldsymbol{\omega}_{bt,k-1}$ and $\bar{\boldsymbol{\omega}} \triangleq (\boldsymbol{\omega}_{k-1} + \boldsymbol{\omega}_k)/2$, and $\mathbf{q}\{\cdot\}$ as defined in (49). Notice that Q0F and Q0B are proper $SO(3)$ integrators: they integrate the angular rates in $\mathfrak{so}(3)$ using forward or backward Euler, producing an angular step $\Delta\boldsymbol{\theta} = \boldsymbol{\omega}\Delta t \in \mathfrak{so}(3)$, and construct the quaternion step $\Delta\mathbf{q} = \mathbf{q}\{\Delta\boldsymbol{\theta}\} = \exp(\Delta\boldsymbol{\theta}/2)$ using the exponential map (49), which is then composed in the $SO(3)$ group locally (*i.e.*, at the right side of the product). Q1 accounts for second-order terms appearing only when the rotation axis changes direction within the integration interval (*i.e.*, $\boldsymbol{\omega}_{k-1}\times\boldsymbol{\omega}_k \neq 0$).

Integrations for the ESKF and the EKF are detailed in the following paragraphs. For the sake of clarity, in this section we integrate the kinematic equations using only $\mathbf{F}_1$ (also known as backward Euler integration) for all variables except the quaternion. For the quaternion, we use Q0B. This choice is pertinent: as will be revealed in the benchmarking, improving the approximations of the transition matrix and the quaternion beyond the forms presented in this section has only minimal effect on the overall performance.

ESKF: For the ESKF we need to integrate the nominal- and the error-state equations. The integration of the nominal-state equations (17) results in

$$\mathbf{p} \leftarrow \mathbf{p} + \mathbf{v}\,\Delta t \tag{22a}$$
$$\mathbf{v} \leftarrow \mathbf{v} + (\mathbf{R}\,(\mathbf{a}_S - \mathbf{a}_b) + \mathbf{g})\,\Delta t \tag{22b}$$
$$\mathbf{q} \leftarrow \mathbf{q} \otimes \mathbf{q}\{(\boldsymbol{\omega}_S - \boldsymbol{\omega}_b)\Delta t\} \tag{22c}$$
$$\mathbf{a}_b \leftarrow \mathbf{a}_b \tag{22d}$$
$$\boldsymbol{\omega}_b \leftarrow \boldsymbol{\omega}_b\,, \tag{22e}$$

where "$\leftarrow$" stands for "gets updated with", *i.e.*, $x \leftarrow f(x, \bullet)$ is equivalent to $x_k = f(x_{k-1}, \bullet_k)$. Similarly, the integration of the error-state equations (18) produces, for a globally-defined error GE,

$$\delta\mathbf{p} \leftarrow \delta\mathbf{p} + \delta\mathbf{v}\,\Delta t \tag{23a}$$
$$\delta\mathbf{v} \leftarrow \delta\mathbf{v} - ([\mathbf{R}\,(\mathbf{a}_S - \mathbf{a}_b)]_\times \delta\boldsymbol{\theta} + \mathbf{R}\delta\mathbf{a}_b)\,\Delta t + \mathbf{v}_i \tag{23b}$$
$$\delta\boldsymbol{\theta} \leftarrow \delta\boldsymbol{\theta} - \mathbf{R}\delta\boldsymbol{\omega}_b\Delta t + \boldsymbol{\theta}_i \tag{23c}$$
$$\delta\mathbf{a}_b \leftarrow \delta\mathbf{a}_b + \mathbf{a}_i \tag{23d}$$
$$\delta\boldsymbol{\omega}_b \leftarrow \delta\boldsymbol{\omega}_b + \boldsymbol{\omega}_i\,, \tag{23e}$$

whereas for a local definition LE,

$$\delta\mathbf{v} \leftarrow \delta\mathbf{v} - (\mathbf{R}[(\mathbf{a}_S - \mathbf{a}_b)]_\times \delta\boldsymbol{\theta} + \mathbf{R}\delta\mathbf{a}_b)\Delta t + \mathbf{v}_i \quad (23f)$$

$$\delta\boldsymbol{\theta} \leftarrow \delta\boldsymbol{\theta} - ([\boldsymbol{\omega}_S - \boldsymbol{\omega}_b]_\times \delta\boldsymbol{\theta} + \delta\boldsymbol{\omega}_b)\Delta t + \boldsymbol{\theta}_i\,. \quad (23g)$$

Here, $\mathbf{v}_i$, $\boldsymbol{\theta}_i$, $\mathbf{a}_i$ and $\boldsymbol{\omega}_i$ are random impulses applied to the velocity, orientation and bias estimates, modeled with Gaussian processes. Their mean is zero, and their covariances matrices are obtained by integrating the variances of the IMU measurement noises, $\mathbf{a}_n$, $\boldsymbol{\omega}_n$, and the IMU bias random walks, $\mathbf{a}_\omega$, $\boldsymbol{\omega}_\omega$, over the time step Δt,

$$\mathbf{V}_i = \mathbf{A}_n\,\Delta t^2 \quad = \sigma_{a_n}^2\,\Delta t^2\,\mathbf{I} \quad [m^2/s^2] \quad (24a)$$

$$\boldsymbol{\Theta}_i = \boldsymbol{\Omega}_n\,\Delta t^2 \quad = \sigma_{\omega_n}^2\,\Delta t^2\,\mathbf{I} \quad [rad^2] \quad (24b)$$

$$\mathbf{A}_i = \mathbf{A}_w\,\Delta t \quad = \sigma_{a_w}^2\,\Delta t\,\mathbf{I} \quad [m^2/s^4] \quad (24c)$$

$$\boldsymbol{\Omega}_i = \boldsymbol{\Omega}_w\,\Delta t \quad = \sigma_{\omega_w}^2\,\Delta t\,\mathbf{I} \quad [rad^2/s^2]\,, \quad (24d)$$

where $\sigma_{a_n}[m/s^2]$, $\sigma_{\omega_n}[rad/s]$, $\sigma_{a_w}[m/s^2\sqrt{s}]$ and $\sigma_{\omega_w}[rad/s\sqrt{s}]$ are to be determined from the information in the IMU datasheet, from real measurements, or –preferably as a last resort– via filter tuning.

EKF: In this case, we simply integrate the true-state kinematic equations (16). Notice that the result is equivalent to the nominal integration in ESKF, but incorporating the noises $\mathbf{v}_i$, $\boldsymbol{\theta}_i$, and biases random walks $\mathbf{a}_i$, $\boldsymbol{\omega}_i$,

$$\mathbf{p}_t \leftarrow \mathbf{p}_t + \mathbf{v}_t\,\Delta t \quad (25a)$$

$$\mathbf{v}_t \leftarrow \mathbf{v}_t + (\mathbf{R}_t\,(\mathbf{a}_S - \mathbf{a}_{bt}) + \mathbf{g})\,\Delta t + \mathbf{v}_i \quad (25b)$$

$$\mathbf{q}_t \leftarrow \mathbf{q}_t \otimes \mathbf{q}\{(\boldsymbol{\omega}_S - \boldsymbol{\omega}_{bt})\Delta t + \boldsymbol{\theta}_i\} \quad (25c)$$

$$\mathbf{a}_{bt} \leftarrow \mathbf{a}_{bt} + \mathbf{a}_i \quad (25d)$$

$$\boldsymbol{\omega}_{bt} \leftarrow \boldsymbol{\omega}_{bt} + \boldsymbol{\omega}_i\,. \quad (25e)$$

2.4 ESKF filter

We are interested in estimating the true-state $\mathbf{x}_t$, which we do as follows. High-frequency IMU data is integrated into the nominal-state $\mathbf{x}$, which does not take into account noise terms or other possible model imperfections and, as a consequence, it accumulates errors. These errors are collected in the error-state, defined as the multivariate Gaussian $\delta\mathbf{x} \sim \mathcal{N}\{\widehat{\delta\mathbf{x}}, \mathbf{P}\}$, this time incorporating all the noise and perturbations. In parallel with integration of the nominal-state, the ESKF predicts a prior estimate of this error-state, and uses the other sensor readings (flow and range) in a correction phase to provide a posterior. After each correction, the error-state's mean ($\widehat{\delta\mathbf{x}}$) is injected into the nominal-state, and then reset to zero. Because of this reset, at each time the nominal state $\mathbf{x}$ is the best estimate of the true state $\mathbf{x}_t$, and the estimated uncertainty is described by the error covariances matrix $\mathbf{P}$.

2.4.1 Prediction

Apart from the true-, nominal- and error-state vectors, it is convenient here to consider our kinematic models in a generic form $\mathbf{x}_t \leftarrow f(\mathbf{x}_t, \mathbf{u}, \mathbf{i})$ that we will identify with the appropriate equation numbers. The input vector $\mathbf{u}$ (IMU readings) and the perturbation impulses vector $\mathbf{i}$ are defined as follows

$$\mathbf{u} = \begin{bmatrix}\mathbf{a}_S \\ \boldsymbol{\omega}_S\end{bmatrix}\,, \quad \mathbf{i} = \begin{bmatrix}\mathbf{v}_i \\ \boldsymbol{\omega}_i \\ \boldsymbol{a}_i \\ \boldsymbol{\omega}_i\end{bmatrix}\,. \quad (26)$$

At the arrival of a new IMU measurement, we propagate the nominal-state $\mathbf{x}$ according to a version of (22) using the selected $\mathbf{F}_N$,

$$\mathbf{x} \leftarrow f(\mathbf{x}, \mathbf{u}, \mathbf{0})\,, \quad (27)$$

and the error-state Gaussian with the filter using (23),

$$\widehat{\delta\mathbf{x}} \leftarrow \mathbf{F}_N\,\widehat{\delta\mathbf{x}} \quad (28a)$$

$$\mathbf{P} \leftarrow \mathbf{F}_N\,\mathbf{P}\,\mathbf{F}_N^\top + \mathbf{F}_i\,\mathbf{Q}_i\,\mathbf{F}_i^\top\,, \quad (28b)$$

where the equation (28a) for $\widehat{\delta\mathbf{x}}$ can be neglected because the error mean $\widehat{\delta\mathbf{x}}$ starts and remains at zero, $\mathbf{F}_N$ is the transition matrix, the Jacobian of (23) with respect to the error-state $\delta\mathbf{x}$ —see App. B for details. $\mathbf{F}_i$ is the Jacobian of (23) with respect to the perturbations vector $\mathbf{i}$, obtained by simple inspection, and $\mathbf{Q}_i$ is the covariances matrix of the perturbation impulses, given by

$$\mathbf{Q}_i = \begin{bmatrix}\mathbf{V}_i & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \boldsymbol{\Theta}_i & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{A}_i & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \boldsymbol{\Omega}_i\end{bmatrix}\,. \quad (29)$$

2.4.2 Innovation and correction

We consider the arrival of sensor data other than IMU, with a model $\mathbf{y} = h_j(\mathbf{x}_t, \mathbf{n}_j)$, with $j = \phi$ for the flow sensor observation model (9,13), and $j = r$ for the range sensor model (14). Because $\widehat{\delta\mathbf{x}} = 0$, we have $\widehat{\mathbf{x}}_t = \mathbf{x}$, and the innovation $\mathbf{z}$ and its covariance $\mathbf{Z}$ read

$$\mathbf{z} = \mathbf{y} - h_j(\mathbf{x}, \mathbf{0}) \quad (30a)$$

$$\mathbf{Z} = \mathbf{H}_j\,\mathbf{P}\,\mathbf{H}_j^\top + \mathbf{N}_j\,, \quad (30b)$$

with $\mathbf{H}_j = \partial h_j/\partial\delta\mathbf{x}$ being the observation Jacobian of flow (9) or range (14), defined with respect to the error-state $\delta\mathbf{x}$. In order to be robust to possible measurement outliers, we perform a χ^2-test based on the Mahalanobis

distance of the innovation (Bar-Shalom et al., 2004). Inliers are validated by checking the condition

$$\mathbf{z}^\top \mathbf{Z}^{-1} \mathbf{z} \leq \chi^2_{th}\,, \tag{31}$$

with χ^2_{th} equal to the 0.95 probability quantile of the χ^2 distribution. If we pass the test, we proceed by computing the Kalman gain $\mathbf{K}$ and observing the filter error,

$$\mathbf{K} = \mathbf{P}\mathbf{H}^\top \mathbf{Z}^{-1} \tag{32a}$$

$$\widehat{\delta\mathbf{x}} \leftarrow \mathbf{K}\mathbf{z} \tag{32b}$$

$$\mathbf{P} \leftarrow \mathbf{P} - \mathbf{K}\,\mathbf{Z}\,\mathbf{K}^\top\,. \tag{32c}$$

We finally update the nominal-state with the observed error mean using the appropriate compositions $\mathbf{x} \leftarrow \mathbf{x} \oplus \widehat{\delta\mathbf{x}}$ introduced in (2). This operation is a simple addition for most variables except for the orientation, which depends on the error representation. Hence, for a global definition (GE) we have

$$\mathbf{q} \leftarrow \mathbf{q}\{\widehat{\delta\boldsymbol{\theta}}\} \otimes \mathbf{q}\,, \tag{33a}$$

whereas for a locally defined error (LE),

$$\mathbf{q} \leftarrow \mathbf{q} \otimes \mathbf{q}\{\widehat{\delta\boldsymbol{\theta}}\}\,. \tag{33b}$$

Notice that these operations constitute proper updates in the $SO(3)$ manifold represented by unit quaternions.

2.5 EKF filter

In this case, the function $f()$ and its Jacobians $\mathbf{F}_N$ and $\mathbf{F}_i$ are drawn from (25). The forms of $f()$ and $\mathbf{F}_N$ depend on the truncation grade we choose —see Appendix B for details. The prediction step is standard EKF,

$$\widehat{\mathbf{x}}_t \leftarrow f(\widehat{\mathbf{x}}_t, \mathbf{u}, \mathbf{0}) \tag{34a}$$

$$\mathbf{P} \leftarrow \mathbf{F}_N\,\mathbf{P}\,\mathbf{F}_N^\top + \mathbf{F}_i\,\mathbf{Q}_i\,\mathbf{F}_i^\top\,. \tag{34b}$$

The innovation is obtained as in ESKF (30), with flow and range observation Jacobians $\mathbf{H}_i = \partial h_i / \partial \mathbf{x}_t$, $i \in \{\phi, r\}$, this time deriving (9) and (14) with respect to the true-state $\mathbf{x}_t$ instead of the error-state. We perform the same outlier rejection explained for the ESKF. Correction follows the standard EKF formulation,

$$\mathbf{K} = \mathbf{P}\mathbf{H}_i^\top \mathbf{Z}^{-1} \tag{35a}$$

$$\widehat{\mathbf{x}}_t \leftarrow \widehat{\mathbf{x}}_t + \mathbf{K}\mathbf{z} \tag{35b}$$

$$\mathbf{P} \leftarrow \mathbf{P} - \mathbf{K}\,\mathbf{Z}\,\mathbf{K}^\top\,. \tag{35c}$$

Notice that, unlike the ESKF updates (33), the sum in (35b) implies that the orientation escapes the $SO(3)$ manifold, and thus that quaternion re-normalization is required.

2.6 Observability analysis

The observability analysis of the system needs the evaluation of the rank and continuous symmetries of the observability matrix defined from the Lie derivatives (Martinelli, 2012). Following this work, we detect three continuous symmetries, corresponding to the non-observable modes of XY translation, and rotation around the direction of gravity (*i.e.*, the *yaw* angle),

$$\boldsymbol{\omega}_s^1 = [1, 0, 0, 0, \cdots, 0]$$

$$\boldsymbol{\omega}_s^2 = [0, 1, 0, 0, \cdots, 0]$$

$$\boldsymbol{\omega}_s^3 = [-p_y, p_x, 0, -v_y, v_x, 0, -\frac{q_z}{2}, -\frac{q_y}{2}, \frac{q_x}{2}, \frac{q_w}{2}, 0, \cdots]\,,$$

where $\{p_x, p_y, v_x, v_y, q_x, q_y, q_z, q_w\}$ are position, velocity and quaternion components. All other modes, including all biases, are observable as long as the maneuvers performed span the observable directions. As shown in (Santamaria-Navarro et al., 2015), the limitations on maneuverability imposed by the MAV dynamics have a negative impact on the observability of certain modes, in particular on the accelerometer bias in the XY axes, and the gyrometer bias in the Z axis. These biases are observable only when the MAV escapes the hovering attitude, and their convergence increases the further we deviate from hovering. Therefore, it is beneficial to drive the MAV in aggressive maneuvers. This requires a flight controller with good stability conditions away from the hovering situation, such as the one we present hereafter.

3 Platform Control

Our platform of choice is a quadrotor due to its mechanical simplicity (Michael et al., 2010) and ease of control. The ability of a quadrotor to operate in confined spaces, hover in space and perch or land on a flat surfaces makes it a very attractive aerial platform. This section describes the dynamic model and the control scheme.

3.1 Dynamic model

Quadrotors are typically equipped with four aligned coplanar propellers. Motion control is achieved by altering the rotation speed of these propellers, thereby changing its torque load and thrust lift characteristics (see Fig. 5).

Let us consider a global coordinate frame w, assumed inertial and defined by unitary column vectors $[{}^w\mathbf{x}, {}^w\mathbf{y}, {}^w\mathbf{z}]$, and a body reference frame b, defined also by $[{}^b\mathbf{x}, {}^b\mathbf{y}, {}^b\mathbf{z}]$ and centered in the center of mass of the

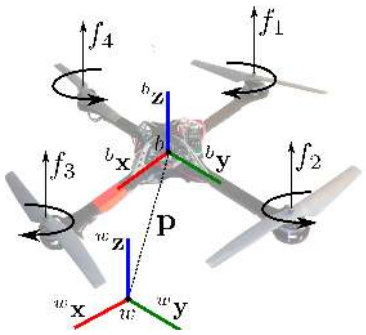


Fig. 5 Quadrotor scheme with reference frames, thrust vectors and propeller rotation directions.

vehicle. The dynamic model of the vehicle can be expressed as

$$\dot{\mathbf{p}} = \mathbf{v} \tag{36a}$$
$$m\,\mathbf{a} = -f\,\mathbf{R}\,{}^w\mathbf{z} + m\,\mathbf{g} \tag{36b}$$
$$\dot{\mathbf{R}} = \mathbf{R}[\boldsymbol{\omega}]_\times \tag{36c}$$
$$\boldsymbol{\mathcal{I}}\dot{\boldsymbol{\omega}} + \boldsymbol{\omega} \times \boldsymbol{\mathcal{I}}\boldsymbol{\omega} = \boldsymbol{\tau}, \tag{36d}$$

where $m \in \mathbb{R}$ is the mass, $\boldsymbol{\mathcal{I}} \in \mathbb{R}^{3\times3}$ is the inertia matrix with respect to the body frame, and ${}^w\mathbf{z} \triangleq \begin{bmatrix}0\ 0\ 1\end{bmatrix}^\top$. The control inputs of the plant are the total thrust $f \in \mathbb{R}$, and the total moment $\boldsymbol{\tau} = \begin{bmatrix}\tau_1\ \tau_2\ \tau_3\end{bmatrix}^\top \in \mathbb{R}^3$ along all axes of the body-fixed frame. The dynamics of rotors and propellers are neglected and it is assumed that the force f_i of each propeller is directly controlled. The total thrust, $f = \sum_{j=1}^{4} f_j$, acts in the direction of the z axis of the body-fixed frame, which is orthogonal to the plane defined by the centers of the four propellers. The relationship between the single motor forces f_i, the total thrust f, and the total moment $\boldsymbol{\tau}$, can be written as

$$\begin{bmatrix} f \\ \tau_1 \\ \tau_2 \\ \tau_3 \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 0 & -d & 0 & d \\ d & 0 & -d & 0 \\ -c & c & -c & c \end{bmatrix} \begin{bmatrix} f_1 \\ f_2 \\ f_3 \\ f_4 \end{bmatrix}, \tag{37}$$

where c is a constant value and d is the distance from the center of mass (b) to a rotor axis, considering all rotors equidistant. For non-zero values of d, (37) can be inverted, therefore our assumption that f and $\boldsymbol{\tau}$ are the inputs of the plant is valid.

3.2 Position and attitude controllers

We want to control the quadrotor with desired positions, heading, linear velocities and accelerations (*i.e.*, $\mathbf{p}_d$, ψ_d, $\mathbf{v}_d$ and $\mathbf{a}_d$) with a controller design based on the nonlinear tracking controller developed on the special Euclidean group $SE(3)$ (Lee et al., 2013). For this, the quadrotor control inputs f, $\boldsymbol{\tau}$ from (37) (see Fig. 2) are chosen as

$$f = -\left(-k_p\mathbf{p}_e - k_v\mathbf{v}_e - m\,\mathbf{g} + m\mathbf{a}_d\right) \cdot \mathbf{R}\,{}^w\mathbf{z} \tag{38a}$$
$$\begin{aligned}\boldsymbol{\tau} = &-k_\theta\boldsymbol{\theta}_e - k_\omega\boldsymbol{\omega}_e + \boldsymbol{\omega}\times\boldsymbol{\mathcal{I}}\boldsymbol{\omega} \\ &- \boldsymbol{\mathcal{I}}\left([\boldsymbol{\omega}]_\times\mathbf{R}^\top\mathbf{R}_C\boldsymbol{\omega}_C - \mathbf{R}^\top\mathbf{R}_C\dot{\boldsymbol{\omega}}_C\right),\end{aligned} \tag{38b}$$

with k_p, k_v, k_θ, k_ω positive definite gains to be tuned. $\mathbf{p}_e$, $\mathbf{v}_e$, $\boldsymbol{\theta}_e$ and $\boldsymbol{\omega}_e$ are the position, velocity, orientation and angular rate errors, defined by

$$\mathbf{p}_e = \mathbf{p} - \mathbf{p}_d \tag{39a}$$
$$\mathbf{v}_e = \mathbf{v} - \mathbf{v}_d \tag{39b}$$
$$\boldsymbol{\theta}_e = \frac{1}{2}\left[\mathbf{R}_C^\top\mathbf{R} - \mathbf{R}^\top\mathbf{R}_C\right]^\times \tag{39c}$$
$$\boldsymbol{\omega}_e = \boldsymbol{\omega} - \mathbf{R}^\top\mathbf{R}_C\boldsymbol{\omega}_C. \tag{39d}$$

$\mathbf{R}_C$ and $\boldsymbol{\omega}_C$ are the internally controlled orientation and angular velocity, as produced by the position controller, refer to (Lee et al., 2013) for more details on their definitions. The symbol $[\cdot]^\times$ represents the map $\mathfrak{so}(3) \to \mathbb{R}^3$, which is the inverse operation of $[\cdot]_\times$.

Using this controller, if the initial attitude error is less than 90°, the zero equilibrium of the tracking errors is exponentially stable, *i.e.*, $\begin{bmatrix}\mathbf{p}_e\ \mathbf{v}_e\ \boldsymbol{\theta}_e\ \boldsymbol{\omega}_e\end{bmatrix} \to \mathbf{0}$. Furthermore, if the initial attitude error is between 90° and 180°, then the zero equilibrium of the tracking errors is almost globally exponentially attractive. The reader can refer to (Lee et al., 2013) for convergence and stability analysis and to (Mellinger and Kumar, 2011) for experimental results.

3.3 Trajectory planning

With the planning module (see Fig 2) we generate trajectories in Cartesian space. These trajectories consist of the desired values fed to the controller above, $\mathbf{p}_d$, ψ_d, $\mathbf{v}_d$ and $\mathbf{a}_d$. Our planner design is based on (Mellinger and Kumar, 2011) which guarantees dynamically feasible trajectories by proving that our dynamic system (36) is differential flat (Fliess et al., 1995). This means that our dynamic system can be formulated as an algebraic function of the flat outputs, which are

$$\eta = \begin{bmatrix}\mathbf{p}^\top\ \psi\end{bmatrix}, \tag{40}$$

or their derivatives. These algebraic relations involve the fourth derivative of the position $\mathbf{p}$, called snap, and the second derivative of the heading ψ (Mellinger and Kumar, 2011). Therefore, to generate smooth and feasible 3D trajectories, it is convenient and sufficient to minimize this snap using the following cost functional:

$$\min \int_{t_0}^{t_f} \left(\mu_p \left\| \frac{\partial^4\mathbf{p}(t)}{\partial t^4} \right\|^2 + \mu_\psi \left\| \frac{\partial^2\psi(t)}{\partial t^2} \right\|^2\right) dt \tag{41}$$

where μ_p and μ_ψ are tuning parameters, subject to the desired boundary conditions on the flat outputs and their concerned derivatives. This minimization problem can be formulated as a quadratic program (Mellinger and Kumar, 2011), also including intermediate waypoints.

4 Validation

We validate our method comparing the produced estimates with respect to precise ground truth measurements. Notice that we do not compare our performances against the more sophisticated VIO algorithms for the reasons exposed in the introduction, namely, the lack of key-frames and lengthy feature tracks in our estimation pipeline. To properly analyze estimation drift, in all experiments, state estimates are initialized with ground-truth values.

4.1 Simulation results

In order to study the performances and limitations of the proposed state estimation setup, we first present experiments with synthetic data under realistic flight conditions. We benchmark all filter types using the same scenario and with the quadrotor simulation equipped with an IMU, a smart camera and a range sensor, and taking advantage of a MATLAB toolbox (checkout our online software [3] for more details on quadrotor dynamic values and sensor parameters). For the benefit of the community, we also make the MATLAB odometry estimation code public[4]. The optimized high-rate C++ implementation is available upon request.

4.1.1 Position RMSE and orientation error evaluation

To analyze the resulting filter estimations we perform N trajectory simulations. We evaluate the Root Mean Square Error (RMSE) between each component i of the estimated vehicle positions (x,y,z) with respect to ground truth, for all time steps k

$$\epsilon_i = \sqrt{\frac{1}{N\,s}\sum_{j=1}^{N}\sum_{k=1}^{s}(p_{i,k} - \hat{p}_{i,k}^{j})^2}\,, \tag{42}$$

where s is the number of time samples of each experiment, $p_{i,k}$ is the i-th component of the true vehicle position at time k, and $\hat{p}_{i,k}^{j}$ is its estimate mean, computed by the filter, corresponding to the j-th among N simulated trajectories.

[3] https://gitlab.iri.upc.edu/asantamaria/QuadSim
[4] https://gitlab.iri.upc.edu/asantamaria/QuadOdom

To analyze the orientation error we use as in (Loianno et al., 2015a), which in turn is based on (Bullo and Lewis, 2004), the orientation error metric defined as

$$\Psi = \frac{1}{2}tr(\mathbf{I} - \mathbf{R}^\top\widehat{\mathbf{R}}) \quad \in \mathbb{R}\,, \tag{43}$$

where $\mathbf{R}$ and $\widehat{\mathbf{R}}$ are respectively the ground truth and estimated vehicle orientations.

Table 2 shows both position RMSE and the abovementioned orientation error metric (no units) achieved at the end of $N = 20$ simulated flights of almost 10 min and 500 m each, performing representative movements (*e.g.* up/down, forward/backward, left/right). For the sake of simplicity only some of the filter variants are reported, and to ease the comparison some filter characteristics are colored. The results in Table 2 show that there is no significant performance difference between filter designs. Moreover, note that all filter types have practically the same computation load as their main differences are not in terms of computation (CPU ticks) but in complexity on their developments and definitions. Being more specific, the computation difference between an ESKF and EKF are the quaternion renormalization required in EKF and the extra composition of error- and nominal- states for the ESKF (2), resulting in minimal operations with similar computation. The extra elements of the Taylor series expansions when using different grades of truncations for the transition matrices ($\mathbf{F}_1$, $\mathbf{F}_2$, $\mathbf{F}_3$) increase the computation time with negligible extra CPU ticks. The different quaternion integration methods (Q0F, Q0B, Q1) entail the same operations except for the Q1 integration that has an extra sum, that again its computation load can be considered negligible.

4.1.2 Average NEES evaluation

A recursive estimator is consistent when the estimation errors are zero-mean and have covariance matrix equal to that reported by the estimator. To evaluate the consistency of the filters we use, as in (Solà et al., 2011) which in turn is based in (Bar-Shalom et al., 2004), the Average Normalized Estimation Error Squared (ANEES) for N Monte Carlo runs, defined as

$$\eta_k = \frac{1}{N}\sum_{j=1}^{N}\left(\mathbf{B}_k - \widehat{\mathbf{B}}_k^j\right)^\top \mathbf{P}_k^{j\,-1}\left(\mathbf{B}_k - \widehat{\mathbf{B}}_k^j\right)\,, \tag{44}$$

where $\mathbf{B}_k$ is the 6 DOF true body pose at time k (*i.e.*, ground truth) and $\mathcal{N}\{\widehat{\mathbf{B}}_k^j, \mathbf{P}_k^j\}$ is its Gaussian estimate, obtained by filtering, corresponding to the j-th among

Error Component ϵ_i	Filter Variant										
	EKF $\mathbf{F}_1$ Q0F LE	EKF $\mathbf{F}_1$ Q0B LE	EKF $\mathbf{F}_1$ Q1 LE	EKF $\mathbf{F}_2$ Q1 LE	EKF $\mathbf{F}_3$ Q1 LE	ESKF $\mathbf{F}_1$ Q0F GE	ESKF $\mathbf{F}_1$ Q0B GE	ESKF $\mathbf{F}_1$ Q1 GE	ESKF $\mathbf{F}_2$ Q1 GE	ESKF $\mathbf{F}_3$ Q1 GE	**ESKF** $\mathbf{F}_1$ Q0F LE
x (m)	10.54	10.48	10.30	10.26	10.26	10.58	10.37	10.13	10.12	10.12	10.38
y (m)	11.13	11.07	10.85	10.81	10.81	11.00	10.82	10.55	10.58	10.58	10.91
z (mm)	7	6	7	6	6	7	7	7	7	7	7
Ψ ($\cdot 10^{-3}$)	2	2	2	2	2	2	2	2	2	2	2

Table 2 Estimation error statistics after 10 min flights of 500 m in straight line. Root Mean Squared Error (RMSE) over 20 experiments for Cartesian position elements (x, y, z) and rotation error index (Ψ) at the end of the trajectory. Color in the filter variant names are added for comparison purposes (those variants with the same color only differ from the colored characteristic).

the N Monte Carlo runs. Each run is done with a different seed for the random generator affecting the process noises and the measurement noises. We now can compute the double-sided 95% probability concentration region, which, for 6 DOF and $N = 25$ runs, has the upper and lower bounds given by

$$\overline{\eta} = \frac{\chi^2_{(25\times 6)}(1-0.975)}{25} = 7.432 \tag{45a}$$

$$\underline{\eta} = \frac{\chi^2_{(25\times 6)}(1-0.025)}{25} = 4.719\,. \tag{45b}$$

If $\eta_k < \underline{\eta}$ for a significant amount of time (more than 2.5% of the time), the filter is considered conservative. Similarly, if $\eta_k > \overline{\eta}$ (also by more than 2.5% of the time), the filter is considered optimistic and therefore inconsistent. Fig. 6 shows an example of the ANEES for the 6 DOF body frame pose $\left[x\ y\ z\ \phi\ \theta\ \psi\right]^\top$ over 25 runs of the same experiment ($N = 25$). We estimated the pose using the two extreme filter variants in terms of simplicity, an EKF with $\mathbf{F}_1$ and Q0B options; and an ESKF with GE, $\mathbf{F}_3$ and Q1 options. The gray horizontal band between abscissas mark the 95% consistency region with $\overline{\eta} = 4.719$ and $\underline{\eta} = 7.432$. Both filter variants are shown to be neither conservative nor inconsistent (see online software simulator for all involved parameters during simulation and estimation).

4.2 Experimental results

The quadrotor used in the real experiments is the ASCTEC Hummingbird research platform shown in Fig. 1. This platform has an off-the-shelf built-in IMU running at 100 Hz, and we equipped it with a PX4-Flow smart camera (Honegger et al., 2013), with a rate of 200 Hz, and a TeraRangeOne range sensor (Ruffo et al., 2014) with a frequency of up to 800 Hz. The smart camera and the range sensor have a cost of around 100€ each. Note that the specialized PX4-Flow module can be replaced by a comercial camera with similar hardware characteristics (*i.e.*, 16*mm* M12 lens with a pixel binning and subsampling resulting in a 64×64 pixels of resolution) together with the method described in Honegger et al. (2013) programmed in the main CPU, as this specialized module uses simple processor operations: sum of absolute differences (SAD) block matching to compute the optical flow. Note, however, that an increment in the estimation drift is possible should the flow rate decrease when replacing the smart camera with another device with slower frame rate.

The algorithms for odometry estimation and control are running onboard, in an Odroid-XU3 platform (using one of its four CPU cores) with Ubuntu 14.04LTS and ROS Indigo. All experiments have been performed at PERCH lab (Penn Engineering Research Collaborative Hub) indoor testbed, at the University of Pennsylvania, equipped with a Qualisys[5] motion capture system running at 100Hz and used for ground-truth comparison. Some of the lab experiments presented herafter are also shown in the accompanying video.

The first set of experiments consists on executing autonomously several trajectories (*i.e.*, the control part uses only the state estimation as input) and includes take-off and landing maneuvers. Fig. 7(a) and 7(b) show the on-board state estimates compared to Qualisys system measurements for both positioning and orientation in a sample experiment. As detailed in previous sections, the height of the platform (*i.e.*, z axis in Fig. 7(a)) is observable thanks to the range measurement, thus its error is low. Similarly, roll and pitch estimation errors are low due to the observability of the gravity direction provided by the fused IMU data. Fi-

[5] www.qualisys.com

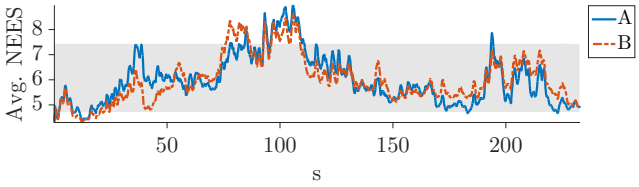
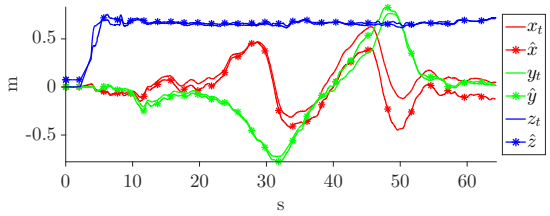
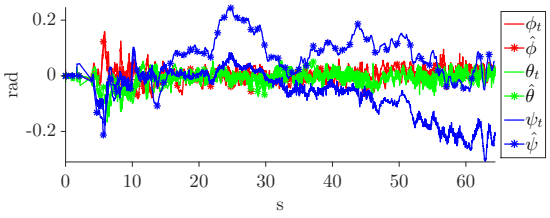


Fig. 6 Average Normalized Estimation Error (ANEES) of the 6 DOF body frame pose $[x\ y\ z\ \phi\ \theta\ \psi]^\top$ over 25 runs of the same experiment. The filter variants are: A) EKF with $\mathbf{F}_1$ and Q0B; and B) ESKF with GE, $\mathbf{F}_3$ and Q1 options. Note how the ANEES is normalized and does not need to differentiate between position and orientation. The gray horizontal band between abscissas $\overline{\eta} = 4.719$ and $\underline{\eta} = 7.432$ mark the 95% consistency region. Both filter propagation and updates are running at 100Hz.



(a) Position



(b) Orientation

Fig. 7 Comparison between the estimation of a sample trajectory (using an ESKF with GE, $\mathbf{F}_3$ and Q1) and ground-truth (Qualisys motion capture system). Ground-truth and estimation variables are labeled as $\cdot_t$ and $\hat{\cdot}$, respectively. The corresponding RMSE is [0.130,0.051,0.094] and the error STD is [0.087,0.050,0.032].

nally, the XY errors grow with time, partly because of the integration of noisy XY velocities, but mostly due to the effect that an unobserved yaw angle ψ has on translation errors.

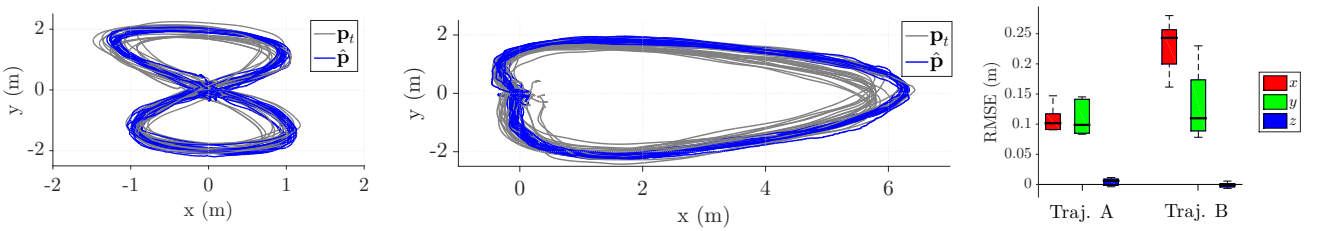
Fig. 8 shows experiments for two different trajectories, 8(a) and 8(b). We launched 25 autonomous runs for each trajectory with a desired height of 1 m and maximum cruise velocity around 1 m/s (notice the superposition of the estimated and ground-truth trajectories in blue and gray respectively). The error statistics for all runs in terms of RMSE are shown in Fig. 8(c). Using similar trajectories we also pushed the smart camera to its limits, by increasing the maximum cruise velocity, and we reached 2.5 m/s flying at 1.5 m height without significant increase in the resulting estimation and control performance.

In order to show the viability of the proposed methods to drive autonomously the vehicle during realistic flight durations, we performed long experiments consisting on continuous trajectory loops during almost 10 min (*i.e.*, a full battery discharge). Fig. 9 shows a comparison between the estimated ($\hat{\mathbf{p}}$) and ground-truth ($\mathbf{p}_t$, obtained with a Qualisys motion capture system) trajectories for one of these experiments with a position RMSE of (0.47 0.67 0.035) (m), and standard deviation (0.29 0.48 0.003) (m). The maximum position error at the end of the flight is (0.73 1.65 0.028) (m). Note that the estimated state (blue in Fig. 9) is used to control the vehicle, thus the estimation errors are reflected in the plot of the ground-truth trajectory (gray in Fig. 9). Although the presented approaches are sufficient to drive autonomously the platform during some minutes without big trajectory errors, as stated before, the x and y positions and yaw angle are not observable (*i.e.*, the method is an odometer) and their output is the result of an incremental estimation subject to drift.

5 Conclusions

In this work, we presented a state estimator design for MAVs that combines low cost and high rate visual-inertial-range sensors. We investigated a wide range of algorithm variations with different computing and implementation complexities. We have shown the feasibility of using such low-cost sensor setup with light algorithms to achieve not only hovering maneuvers but also fully autonomous navigation. All the technical details have been provided, facilitating the use of the proposed methods by other groups in the community.

The result of our experimentation shows that the effects of all the variations in the estimator design are minimal. In particular, the refinements on the transition matrices $\mathbf{F}_1 \cdots \mathbf{F}_3$ have no conclusive effect, meaning that the classical Euler approximation $\mathbf{F}_1$ is sufficiently good. A similar conclusion can be drawn for the quaternion integrators Q0B, Q0F and Q1, and even for the error compositions LE and GE. We conclude that the final choices can be driven more by a criterion of convenience rather than performance. This is due to the high frequency of the measurements and filter updates, which renders all integration schemes close to the continuous-time case, and therefore equivalent in practice. Regarding the filter type, EKF vs. ESKF, we also found equivalent performances. We can base our choice on different criteria. For example, EKF is more widely known, and it is also simpler, both conceptually and in terms of implementation complexity. However, ESKF



(a) Trajectory A. (b) Trajectory B. (c) Positioning RMSE statistics.

Fig. 8 Error analysis of two trajectories with 25 runs each. All runs are executed fully autonomously with a maximum cruise velocity of 1 m/s (best seen in color).

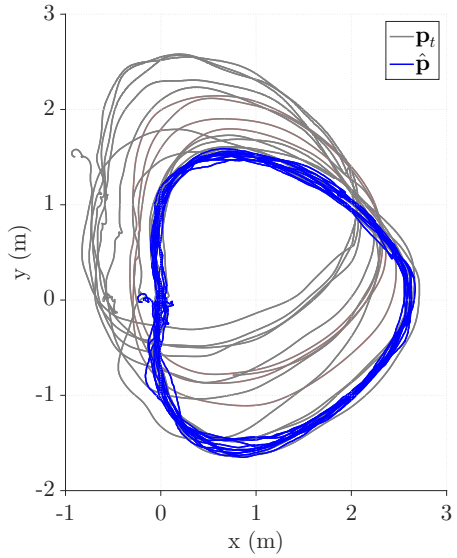


Fig. 9 Position estimation results for a long experiment (almost 10 min of continuous flight and a full battery discharge). Note that in full autonomous mode the vehicle is controlled using the estimation, thus the drift of the platform can be seen in the ground-truth trajectory (Qualisys motion capture system).

is very close to it, and constitutes a more proper and elegant solution, from a theoretical viewpoint, because of its operation in the rotations manifold $SO(3)$. This implies, for example, that in ESKF there is no need to perform quaternion re-normalization. Our final recommendations are the classical EKF with $\mathbf{F}_1$, Q0B and quaternion re-normalization; or the more proper ESKF with $\mathbf{F}_1$, Q0B, and either GE or LE. As both methods require similar number of mathematical operations, both have essentially the same computational cost.

Using these filters, in terms of overall precision, our state estimates are usable during flight times of several minutes, enabling the MAV to perform a number of tasks that require navigation without the aid on any external positioning system.

The estimated state is richer than just odometry, and includes higher derivatives such as velocities and accelerations, all precisely referenced to the gravity direction. These are exploited by a non-linear controller to drive the vehicle in 3D space, showing that the employed sensors are more than sufficient to provide autonomy to an aerial platform. This is the first time that such inexpensive sensors enable precise localization and autonomous navigation of aerial vehicles.

Appendix A Quaternion Conventions and Properties

We use, as in (Solà, 2015), the Hamilton convention for quaternions. If we denote a quaternion ${}^G\mathbf{q}_L$ representing the orientation of a local frame L with respect to a global frame G, then a generic composition of two quaternions is defined as

$$ {}^G\mathbf{q}_C = {}^G\mathbf{q}_L \otimes {}^L\mathbf{q}_C = {}^G\mathbf{Q}_L^+ \, {}^L\mathbf{q}_C = {}^L\mathbf{Q}_C^- \, {}^G\mathbf{q}_L \,, \quad (46) $$

where, for a quaternion $\mathbf{q} = [w, x, y, z]^\top$, we can define $\mathbf{Q}^+$ and $\mathbf{Q}^-$ respectively as the left- and right- quaternion product matrices,

$$ \mathbf{Q}^+ = \begin{bmatrix} w & -x & -y & -z \\ x & w & -z & y \\ y & z & w & -x \\ z & -y & x & w \end{bmatrix} , \quad \mathbf{Q}^- = \begin{bmatrix} w & -x & -y & -z \\ x & w & z & -y \\ y & -z & w & x \\ z & y & -x & w \end{bmatrix} . \quad (47) $$

In the quaternion product, we notice how the right-hand quaternion is defined locally in the frame L, which is specified by the left-hand quaternion. Vector transformation from a local frame L to the global G is performed by the double product

$$ {}^G\mathbf{v} = {}^G\mathbf{q}_L \otimes {}^L\mathbf{v} \otimes ({}^G\mathbf{q}_L)^* = {}^G\mathbf{q}_L \otimes {}^L\mathbf{v} \otimes {}^L\mathbf{q}_G \,, \quad (48) $$

where we use the shortcut $\mathbf{q} \otimes \mathbf{v} \equiv \mathbf{q} \otimes [0, \mathbf{v}]^\top$ for convenience of notation.

Throughout the paper, we note $\mathbf{q}\{x\}$ the quaternion and $\mathbf{R}\{x\}$ the rotation matrix equivalents to a generic orientation x. A rotation $\boldsymbol{\theta} = \theta\mathbf{u}$, of θ radians around

the unit axis $\mathbf{u}$, can be expressed in quaternion and matrix forms using the exponential maps

$$\mathbf{q}\{\boldsymbol{\theta}\} = e^{\boldsymbol{\theta}/2} = \begin{bmatrix} \cos(\theta/2) \\ \mathbf{u}\sin(\theta/2) \end{bmatrix} \xrightarrow[\theta\to 0]{} \begin{bmatrix} 1 \\ \boldsymbol{\theta}/2 \end{bmatrix}, \tag{49}$$

$$\begin{aligned} \mathbf{R}\{\boldsymbol{\theta}\} &= e^{[\boldsymbol{\theta}]_\times} = \mathbf{I} + \sin\theta[\mathbf{u}]_\times + (1-\cos\theta)[\mathbf{u}]_\times^2 \\ &\xrightarrow[\theta\to 0]{} \mathbf{I} + [\boldsymbol{\theta}]_\times \end{aligned} \tag{50}$$

We also write $\mathbf{R} = \mathbf{R}\{\mathbf{q}\}$, according to

$$\mathbf{R}\{\mathbf{q}\} = \begin{bmatrix} w^2+x^2-y^2-z^2 & 2(xy-wz) & 2(xz+wy) \\ 2(xy+wz) & w^2-x^2+y^2-z^2 & 2(yz-wx) \\ 2(xz-wy) & 2(yz+wx) & w^2-x^2-y^2+z^2 \end{bmatrix} \tag{51}$$

Finally, the time-derivative of the quaternion is

$$\dot{\mathbf{q}} = \frac{1}{2}\boldsymbol{\Omega}(\boldsymbol{\omega})\mathbf{q} = \frac{1}{2}\mathbf{q}\otimes\boldsymbol{\omega}\,, \tag{52}$$

with $\boldsymbol{\omega}$ the angular velocity in body frame, and $\boldsymbol{\Omega}$ the skew-symmetric matrix defined as

$$\boldsymbol{\Omega}(\boldsymbol{\omega}) \triangleq \mathbf{Q}^-(\boldsymbol{\omega}) = \begin{bmatrix} 0 & -\boldsymbol{\omega}^\top \\ \boldsymbol{\omega} & -[\boldsymbol{\omega}]_\times \end{bmatrix}. \tag{53}$$

Appendix B Filter Transition Matrices

We detail the construction of the filter transition matrix for the three involved integrals: ESKF nominal- (22), ESKF error- (23), and EKF true- (25) kinematics. For each case, we need to define the matrix $\mathbf{A}$ as the Jacobian of the respective continuous-time system, and build the transition matrix $\mathbf{F}_N$ as the truncated Taylor series (20), *i.e.*,

$$\mathbf{F}_N = \sum_{n=0}^{N} \frac{1}{n!}\mathbf{A}^n\Delta t^n = \mathbf{I} + \mathbf{A}\Delta t + \frac{1}{2!}\mathbf{A}^2\Delta t^2 + \cdots$$

In the following paragraphs, we detail the matrices $\mathbf{A}$ for each case, and some examples of their first powers up to $n=3$. The reader should find no difficulties in building the powers of $\mathbf{A}$ that have not been detailed, and the transition matrices $\mathbf{F}_N$ using the Taylor series above.

The Jacobian $\mathbf{A} = \partial f(\mathbf{x}, \delta\mathbf{x}, \cdot)/\partial\delta\mathbf{x}$ of the ESKF's continuous time error-state system $f()$ (18) using GE is,

$$\mathbf{A} = \begin{bmatrix} 0 & \mathbf{I} & 0 & 0 & 0 \\ 0 & 0 & \mathbf{V} & -\mathbf{R} & 0 \\ 0 & 0 & 0 & 0 & -\mathbf{R} \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}, \tag{54}$$

with $\mathbf{V} = -[\mathbf{R}(\mathbf{a}_S - \mathbf{a}_b)]_\times$. Its powers are,

$$\mathbf{A}^2 = \begin{bmatrix} 0 & 0 & \mathbf{V} & -\mathbf{R} & 0 \\ 0 & 0 & 0 & 0 & -\mathbf{V}\mathbf{R} \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}, \; \mathbf{A}^3 = \begin{bmatrix} 0 & 0 & 0 & 0 & -\mathbf{V}\mathbf{R} \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix},$$

and $\mathbf{A}^n = \mathbf{0}$ for $n > 3$. For LE we have

$$\mathbf{A} = \begin{bmatrix} 0 & \mathbf{I} & 0 & 0 & 0 \\ 0 & 0 & \mathbf{V} & -\mathbf{R} & 0 \\ 0 & 0 & \boldsymbol{\Theta} & 0 & -\mathbf{I} \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}, \; \mathbf{A}^2 = \begin{bmatrix} 0 & 0 & \mathbf{V} & -\mathbf{R} & 0 \\ 0 & 0 & \mathbf{V}\boldsymbol{\Theta} & 0 & -\mathbf{V} \\ 0 & 0 & \boldsymbol{\Theta}^2 & 0 & -\boldsymbol{\Theta} \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}, \cdots$$

with $\mathbf{V} = -\mathbf{R}[\mathbf{a}_S - \mathbf{a}_b]_\times$, and $\boldsymbol{\Theta} = -[\boldsymbol{\omega}_S - \boldsymbol{\omega}_b]_\times$.

The Jacobians $\mathbf{A} = \partial f(\mathbf{x}, \cdot)/\partial\mathbf{x}$ of the continuous-time EKF true- (16) and ESKF nominal- (17) systems are equal to each other, having

$$\mathbf{A} = \begin{bmatrix} 0 & \mathbf{I} & 0 & 0 & 0 \\ 0 & 0 & \mathbf{V} & -\mathbf{R} & 0 \\ 0 & 0 & \mathbf{W} & 0 & \mathbf{Q} \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}, \; \mathbf{A}^2 = \begin{bmatrix} 0 & 0 & \mathbf{V} & -\mathbf{R} & 0 \\ 0 & 0 & \mathbf{V}\mathbf{W} & 0 & \mathbf{V}\mathbf{Q} \\ 0 & 0 & \mathbf{W}^2 & 0 & \mathbf{W}\mathbf{Q} \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}, \cdots$$

where $\mathbf{V}$, $\mathbf{W}$ and $\mathbf{Q}$ are defined by

$$\mathbf{V} = \frac{\partial\mathbf{R}\{\mathbf{q}\}\,(\mathbf{a}_S - \mathbf{a}_b)}{\partial\mathbf{q}} \tag{55a}$$

$$\mathbf{W} = \frac{\partial\frac{1}{2}\mathbf{q}\otimes(\boldsymbol{\omega}_S - \boldsymbol{\omega}_b)}{\partial\mathbf{q}} \tag{55b}$$

$$\mathbf{Q} = \frac{\partial\frac{1}{2}\mathbf{q}\otimes(\boldsymbol{\omega}_S - \boldsymbol{\omega}_b)}{\partial\boldsymbol{\omega}_b}, \tag{55c}$$

and are developed hereafter. For the first Jacobian $\mathbf{V}$ it is convenient to recall the derivative of a rotation of a vector $\mathbf{a}$ by a quaternion $\mathbf{q} = [w,x,y,z]^\top = [w,\mathbf{v}]^\top$ with respect to the quaternion,

$$\begin{aligned} \mathbf{V}(\mathbf{q},\mathbf{a}) \triangleq \frac{\partial\mathbf{R}\{\mathbf{q}\}\,\mathbf{a}}{\partial\mathbf{q}} &= \frac{\partial(\mathbf{q}\otimes\mathbf{a}\otimes\mathbf{q}^*)}{\partial\mathbf{q}} \\ &= 2\big[w\mathbf{a}+\mathbf{v}\times\mathbf{a} \;\big|\; \mathbf{v}\mathbf{a}^\top - \mathbf{a}\mathbf{v}^\top + \mathbf{a}^\top\mathbf{v}\mathbf{I}_3 - w[\mathbf{a}]_\times\big], \end{aligned} \tag{56}$$

having therefore

$$\mathbf{V} = \mathbf{V}(\mathbf{q},\, \mathbf{a}_S - \mathbf{a}_b)\,. \tag{57}$$

For the Jacobian $\mathbf{W}$ we have from (52)

$$\mathbf{W} = \frac{1}{2}\boldsymbol{\Omega}(\boldsymbol{\omega}_S - \boldsymbol{\omega}_b)\,, \tag{58}$$

with $\boldsymbol{\Omega}(\boldsymbol{\omega})$ the skew-symmetric matrix defined in (53). Finally, for the Jacobian $\mathbf{Q}$ we use (46), (47) and (49) to obtain

$$\mathbf{Q} = -\frac{1}{2}\begin{bmatrix} -x & -y & -z \\ w & -z & y \\ z & w & -x \\ -y & x & w \end{bmatrix}. \tag{59}$$

References

Bar-Shalom Y., Li X. R., Kirubarajan T. (2004) Estimation with Applications to Tracking and Navigation: Theory Algorithms and Software. John Wiley & Sons.

Blösch M., Weiss S., Scaramuzza D., Siegwart R. (2010) Vision based MAV navigation in unknown and unstructured environments. In: Proc. IEEE Int. Conf. Robotics Autom., Anchorage, pp. 21–28.

Blösch M., Omari S., Fankhauser P., Sommer H., Gehring C., Hwangbo J., Hoepflinger M., Hutter M., Siegwart R. (2014) Fusion of optical flow and inertial measurements for robust egomotion estimation. In: Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst., Chicago, pp. 3102–3107.

Bullo F., Lewis A. (2004) Geometric Control of Mechanical Systems. Springer.

Faessler M., Fontana F., Forster C., Mueggler E., Pizzoli M., Scaramuzza D. (2016) Autonomous, vision-based flight and live dense 3D mapping with a quadrotor micro aerial vehicle. J. Field Robotics 33(4):431–450.

Fliess M., Lévine J., Martin P., Rouchon P. (1995) Flatness and defect of non-linear systems: Introductory theory and examples. Int. J. Control. 61(6):1327–1361.

Forster C., Pizzoli M., Scaramuzza D. (2014) SVO: Fast semi-direct monocular visual odometry. In: Proc. IEEE Int. Conf. Robotics Autom., Hong Kong, pp. 15–22.

Forte F., Naldi R., Marconi L. (2012) Impedance control of an aerial manipulator. In: Proc. Autom. Control Conf., Montreal, pp. 3839–3844.

Fraundorfer F., Heng L., Honegger D., Lee G. H., Meier L., Tanskanen P., Pollefeys M. (2012) Vision-based autonomous mapping and exploration using a quadrotor MAV. In: Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst., Vilamoura, pp. 4557–4564.

Hérissé B., Hamel T., Mahony R., Russotto F. X. (2012) Landing a VTOL unmanned aerial vehicle on a moving platform using optical flow. IEEE Trans. Robotics 28(1):77–89.

Hesch J. A., Kottas D. G., Bowman S. L., Roumeliotis S. I. (2014) Camera-IMU-based localization: Observability analysis and consistency improvement. Int. J. Robotics Res. 33(1):182–201.

Honegger D., Lorenz M., Tanskanen P., Pollefeys M. (2013) An open source and open hardware embedded metric optical flow CMOS camera for indoor and outdoor applications. In: Proc. IEEE Int. Conf. Robotics Autom., Karlsruhe, pp. 1736–1741.

Jones E. S., Soatto S. (2011) Visual-inertial navigation, mapping and localization: A scalable real-time causal approach. Int. J. Robotics Res. 30(4):407–430.

Kelly J., Sukhatme G. S. (2011) Visual-inertial sensor fusion: Localization, mapping and sensor-to-sensor self-calibration. Int. J. Robotics Res. 30(1):56–79.

Lee T., Leok M., McClamroch N. H. (2013) Nonlinear robust tracking control of a quadrotor UAV on SE(3). Asian J. Control 15(2):391–408.

Li M., Mourikis A. I. (2012) Improving the accuracy of EKF-based visual-inertial odometry. In: Proc. IEEE Int. Conf. Robotics Autom., Saint Paul, pp. 828–835.

Li M., Mourikis A. I. (2013) High-precision, consistent EKF-based visual-inertial odometry. Int. J. Robotics Res. 32(6):690–711.

Liu H., Darabi H., Banerjee P., Liu J. (2007) Survey of wireless indoor positioning techniques and systems. IEEE Trans. Syst., Man, and Cyb. 37(6):1067–1080.

Loianno G., Mulgaonkar Y., Brunner C., Ahuja D., Ramanandan A., Chari M., Diaz S., Kumar V. (2015a) Smartphones power flying robots. In: Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst., Hamburg, pp. 1256–1263.

Loianno G., Thomas J., Kumar V. (2015b) Cooperative localization and mapping of MAVs using RGB-D sensors. In: Proc. IEEE Int. Conf. Robotics Autom., Seattle, pp. 4021–4028.

Madyastha V. K., Ravindra V. C., Mallikarjunan S., Goyal A. (2011) Extended Kalman filter vs. error state Kalman filter for aircraft attitude estimation. In: Proc. AIAA Guid., Nav., and Control Conf., Portland, pp. 6615–6638.

Martinelli A. (2012) Vision and IMU data fusion: Closed-form solutions for attitude, speed, absolute scale, and bias determination. IEEE Trans. Robotics 28(1):44–60.

Martinelli A. (2013) Visual-inertial structure from motion: Observability and resolvability. In: Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst., Tokyo, pp. 4235–4242.

Meier L., Tanskanen P., Heng L., Lee G., Fraundorfer F., Pollefeys M. (2012) PIXHAWK: A micro aerial vehicle design for autonomous flight using onboard computer vision. Aut. Robots 33(1-2):21–39.

Mellinger D., Kumar V. (2011) Minimum snap trajectory generation and control for quadrotors. In: Proc. IEEE Int. Conf. Robotics Autom., Shanghai, pp. 2520–2525.

Michael N., Mellinger D., Lindsey Q., Kumar V. (2010) The grasp multiple micro–UAV test bed. IEEE Robotics Autom. Mag. 17(3):56–65.

Michael N., Shen S., Mohta K., Kumar V., Nagatani K., Okada Y., Kiribayashi S., Otake K., Yoshida K.,

Ohno K., Takeuchi E., Tadokoro S. (2012) Collaborative mapping of an earthquake-damaged building via ground and aerial robots. J. Field Robotics 29(5):832–841.

Nikolic J., Rehder J., Burri M., Gohl P., Leutenegger S., Furgale P. T., Siegwart R. (2014) A synchronized visual-inertial sensor system with FPGA preprocessing for accurate real-time SLAM. In: Proc. IEEE Int. Conf. Robotics Autom., Hong Kong, pp. 431–437.

Omari S., Ducard G. (2013) Metric visual-inertial navigation system using single optical flow feature. In: Proc. Eur. Control Conf., Zurich, pp. 1310–1316.

Ozaslan T., Shen S., Mulgaonkar Y., Michael N., Kumar V. (2013) Inspection of penstocks and featureless tunnel-like environments using micro UAVs. In: Proc. Field and Service Robotics Conf., Brisbane, pp. 123–136.

Ravindra V., Madyastha V., Goyal A. (2012) The equivalence between two well-known variants of the Kalman filter. In: Proc. Adv. Cont. Opt. of Dynamic Syst., Bangalore.

Rossi R., Santamaria-Navarro A., Andrade-Cetto J., Rocco P. (2017) Trajectory generation for unmanned aerial manipulators through quadratic programming. IEEE Robotics and Autom. Letters 2(2):389–396.

Roumeliotis S. I., Johnson A. E., Montgomery J. F. (2002) Augmenting inertial navigation with image-based motion estimation. In: Proc. IEEE Int. Conf. Robotics Autom., Washington, vol. 4, pp. 4326–4333.

Roussillon C., Gonzalez A., Solà J., Codol J. M., Mansard N., Lacroix S., Devy M. (2011) RT-SLAM: A generic and real-time visual SLAM implementation. In: Computer Vision Systems, Lect. Notes in Comp. Science, vol. 6962, Springer Berlin Heidelberg, pp. 31–40.

Ruffo M., Di Castro M., Molinari L., Losito R., Masi A., Kovermann J., Rodrigues L. (2014) New infrared time-of-flight measurement sensor for robotic platforms. In: Proc Int. Symp. and Int. Work. on ADC Modelling and Testing, pp. 13–18.

Santamaria-Navarro A., Lippiello V., Andrade-Cetto J. (2014) Task priority control for aerial manipulation. In: Proc. IEEE Int. Symp. Safe. Sec. Resc. Robotics., Toyako-cho, pp. 1–6.

Santamaria-Navarro A., Solà J., Andrade-Cetto J. (2015) High-frequency MAV state estimation using low-cost inertial and optical flow measurement units. In: Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst., Hamburg, pp. 1864–1871.

Santamaria-Navarro A., Grosch P., Lippiello V., Solà J., Andrade-Cetto J. (2017) Uncalibrated visual servo for unmanned aerial manipulation. IEEE/ASME Trans. on Mechatronics 22(4):1610–1621.

Shen S., Michael N., Kumar V. (2012) Autonomous indoor 3D exploration with a micro-aerial vehicle. In: Proc. IEEE Int. Conf. Robotics Autom., Saint Paul, pp. 9–15.

Shen S., Mulgaonkar Y., Michael N., Kumar V. (2013) Vision-based state estimation and trajectory control towards high-speed flight with a quadrotor. In: Proc. Rob. Science and Syst., Berlin.

Solà J. (2015) Quaternion kinematics for the error-state KF, URL https://hal.archives-ouvertes.fr/hal-01122406v5, hal-01122406, v5, in preparation.

Solà J., Vidal-Calleja T., Civera J., Montiel J. M. M. (2011) Impact of landmark parametrization on monocular EKF-SLAM with points and lines. Int. J. Comput. Vision 97(3):339–368.

Thomas J., Loianno G., Sreenath K., Kumar V. (2014) Toward image based visual servoing for aerial grasping and perching. In: Proc. IEEE Int. Conf. Robotics Autom., Hong Kong, pp. 2113–2118.

Tomic T., Schmid K., Lutz P., Domel A., Kassecker M., Mair E., Grixa I., Ruess F., Suppa M., Burschka D. (2012) Toward a fully autonomous UAV: research platform for indoor and outdoor urban search and rescue. IEEE Robotics Autom. Mag. 19(3):46–56.

Trawny N., Roumeliotis S. I. (2005) Indirect Kalman filter for 3D attitude estimation. University of Minnesota, Dept. of Comp. Sci. & Eng., Tech. Rep 2, rev. 57.

Weiss S., Scaramuzza D., Siegwart R. (2011) Monocular-SLAM-based navigation for autonomous micro helicopters in gps denied environments. J. Field Robotics 28(6):854–874.

Weiss S., Achtelik M., Lynen S., Chli M., Siegwart R. (2012) Real-time onboard visual-inertial state estimation and self-calibration of MAVs in unknown environments. In: Proc. IEEE Int. Conf. Robotics Autom., Saint Paul, pp. 957–964.

A. Santamaria-Navarro was born in Moià, Barcelona on October 11, 1982. He received the bachelor degrees in Telecomunications and Industrial management and the Master degree in Automatic Control and Robotics, from the Universitat Politècnica de Catalunya (UPC), Barcelona, Spain, in 2004 and 2012 respectively. He is PhD student at the Institut de Robòtica i Informàtica Industrial, CSIC-UPC, Barcelona, Spain. His current research interests include computer vision, mobile robotics and unmanned aerial vehicles (UAVs).

G. Loianno received B.Sc and M.Sc in Automation Engineering both with honours from the University of Naples Federico II in December 2007 and February 2010, respectively. From August 2008 to March 2009, he has been an exchange student at the Royal Insitute of Technology (KTH) in Stockholm. From September 2009 to March 2010, he developed his master thesis at ETH Zurich at the ASL Laboratory focusing on micro aerial vehicles under the supervision of Dr. Davide Scaramuzza. He received his Ph.D. in computer and control engineering focusing in robotics in May 2014 in the PRISMA Lab group, led by Dr. Bruno Siciliano. He has been involved in the EU FP7 project AIRobots (www.airobots.eu) in sensor fusion and visual control. He has published more than 40 conference papers, journal papers, and book chapters. Beginning in April 2013, he worked for 14 months with the GRASP Lab at the University of Pennsylvania, supervised by Prof. Dr. Vijay Kumar. From June 2014 to July 2015, he was a postdoctoral researcher in his lab at the University of Pennsylvania, where he is currently a research scientist. His research interests include visual odometry, sensor fusion, and visual servoing for micro aerial vehicles.

J. Solà is a Ramón y Cajal (RyC) researcher at IRI-CSIC. He received the B.Sc. degree in telecommunications and electronic engineering from the Universitat Politècnica de Catalunya, in 1995, the M.Sc. degree in control systems from the École Doctorale Systèmes, Toulouse, France, in 2003, and the PhD degree in control systems from the Institut National Polytechnique de Toulouse in 2007, where he was hosted by the Laboratoire d'Analyse et Architecture de Systèmes (LAAS-CNRS). He was a Postdoctoral Fellow at SRI International, Menlo Park, California and at LAAS-CNRS. Joan is an expert in computer vision for robotics, and in particular on monocular SLAM. He has also vast experience on the integration of IMUs for humanoid robotics applications. Joan has also experience as researcher in the private industry. He was developing power converters for renewable energies systems at Ecotècnia s.c.c.l., Barcelona, now Alstom Wind, in the pioneering period of 1995-1999. He has been developing world-wide innovative Litihium-Ion batteries and fault-tolerant power systems for manned deep-water submarines (for more than 1000m depth), at Ictineu Submarins SL, Barcelona. Joan's current interests are in robot perception, localization and mapping, with the aim of closing the loop of perception, through planning, motion generation and control, especially for unstable and agile platforms such as unmanned aerial vehicles and humanoids.

V. Kumar is the UPS Foundation Professor in the School of Engineering and Applied Science at the University of Pennsylvania. He received his Bachelors of Technology from the Indian Institute of Technology, Kanpur and his Ph.D. from The Ohio State University in 1987. He served as the Deputy Dean for Research in the School of Engineering and Applied Science from 2000-2004. He directed the GRASP Laboratory, a multidisciplinary robotics and perception laboratory, from 1998-2004. He was the Chairman of the Department of Mechanical Engineering and Applied Mechanics from 2005-2008 and the Deputy Dean for Education in the School of Engineering and Applied Science from 2008-2012. He was the assistant director for robotics and cyber physical systems at the White House Office of Science and Technology Policy from 2012-2014. Dr. KumarâĂŹs research interests are in robotics, specifically multirobot systems, and micro aerial vehicles. He has served on the editorial boards of the IEEE Transactions on Robotics and Automation, IEEE Transactions on Automation Science and Engineering, ASME Journal of Mechanical Design, the ASME Journal of Mechanisms and Robotics and the Springer Tract in Advanced Robotics (STAR). He is the recipient of the 1991 National Science Foundation Presidential Young Investigator award, the 1996 Lindback Award for Distinguished Teaching (University of Pennsylvania), the 1997 Freudenstein Award for significant accomplishments in mechanisms and robotics, the 2012 ASME Mechanisms and Robotics Award, the 2012 IEEE Robotics and Automation Society Distinguished Service Award and a 2012 World Technology Network Award. He has won best paper awards at DARS 2002, ICRA 2004, ICRA 2011, RSS 2011, and RSS 2013, and has advised doctoral students who have won Best Student Paper Awards at ICRA 2008, RSS 2009, and DARS 2010.

J. Andrade-Cetto (S'94-M'95) received the BSEE degree from CETYS Universidad, Mexico, in 1993, the MSEE degree from Purdue University, USA, in 1995, and the PhD degree in Systems Engineering from the Universitat

Politècnica de Catalunya (UPC), Barcelona, Spain, in 2003. He received the EURON Georges Giralt Best PhD Award in 2005. He is Associate Researcher of the Spanish Scientific Research Council at the Institut de Robòtica i Informàtica Industrial, CSIC-UPC, Barcelona, Spain. His current research interests include state estimation and computer vision with applications to mobile robotics.