Chapter
Odometry Estimation for Aerial Manipulators

A. Santamaria-Navarro, J. Solà, J. Andrade-Cetto

Abstract This chapter explains a fast and low-cost state localization estimation method for small-sized UAVs, that uses an IMU, a smart camera and an infrared time-of-flight range sensor that act as an odometer providing absolute attitude, velocity, orientation, angular rate and acceleration at a rate higher than 100Hz. This allows estimating almost continuously the localization of the aerial robot, when GPS or other methods can at most reach 5Hz. This technique does not require creating a map for localization.

1 Introduction

Combinations of Inertial Measurement Units (IMU) and monocular visual sensors for aerial robot localization 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 aerial robot navigation, the direction of gravity.

The processes of estimating the vehicle state using such sensors are known as Visual-Inertial Odometry (VIO, with no absolute localization) [2, 2], and Visual-Inertial SLAM (enabling absolute localization by revisiting previously mapped areas) [4, 1, 11]. During a aerial manipulation mission, the focus of our work is not at building maps but at localizing the platform, thus we concentrate on VIO. The methods described in this chapter summarize the work presented in [15] and [16].

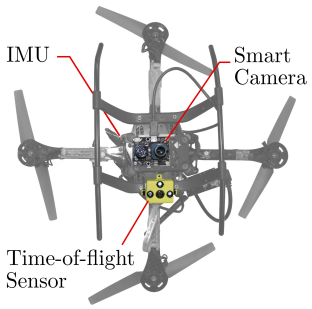


Fig. 1 Bottom view of the ASCTEC Hummingbird quadrotor used in the experiments. The vehicle is equipped with a built-in IMU, a smart camera and a time-of-flight range sensor.

2 Flow-inertial-range odometry

Here, we present a fast and low-cost state estimation method for small-sized UAV. We use exclusively low-cost sensors and low-complexity algorithms. As hardware, we take advantage of a low-cost IMU, a smart camera [3] which directly outputs 2D flow, and an infrared time-of-flight range sensor [12], all shown in Fig. 1. As software, we have developed two Kalman filters, in the extended (EKF) and error-state (ESKF) forms [9], 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 [14, 13, 10].

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 this 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 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 aerial robot 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 [18]. In this work, we use a nonlinear controller that operates

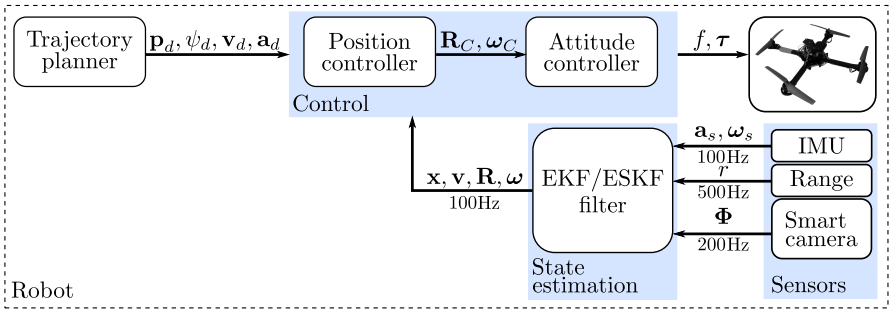


Fig. 2 Overview of the architecture pipeline for estimation and control

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

on the special Euclidean group $SE(3)$, based on the work [5]. 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.

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.

3 EKF and ESKF for odometry estimation

The algorithm variations that we investigate are shown in Table 1, and are properly defined later in [16]. They are summarized hereafter, together with the key works that defended them. First, we implement error-state (ESKF) and extended (EKF) Kalman filters [8]. Second, we express the orientation errors of ESKF both in local (LE) and global (GE) frames [6].[1] Third, we compare different schemes for rotational motion integration of the quaternion [17], 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 ($\mathbf{F}_1, \mathbf{F}_2, \mathbf{F}_3$) [7].

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

Table 2 Symbols used in the development of the flow-inertial-range odometry estimation approach.

Definition	Symbol
"True" state: $\mathbf{x}_t = \begin{bmatrix}\mathbf{p}_t\ \mathbf{v}_t\ \mathbf{q}_t\ \mathbf{a}_{bt}\ \boldsymbol{\omega}_{bt}\end{bmatrix}^\top \in \mathbb{R}^{16}$	$\mathbf{x}_t$
Nominal state: $\mathbf{x} = \begin{bmatrix}\mathbf{p}\ \mathbf{v}\ \mathbf{q}\ \mathbf{a}_b\ \boldsymbol{\omega}_b\end{bmatrix}^\top \in \mathbb{R}^{16}$	$\mathbf{x}$
Error state: $\delta\mathbf{x} = \begin{bmatrix}\delta\mathbf{p}\ \delta\mathbf{v}\ \delta\theta\ \delta\mathbf{a}_b\ \delta\boldsymbol{\omega}_b\end{bmatrix}^\top \in \mathbb{R}^{15}$	$\delta\mathbf{x}$
Accelerometers biases (with true, nominal and error state variants)	$\mathbf{a}_b$
Gyroscope biases (with true, nominal and error state variants)	$\boldsymbol{\omega}_b$
Transition matrix approximations	$\mathbf{F}_1, \mathbf{F}_2, \mathbf{F}_3$

In order to describe the state estimation formulation for ESKF and EKF, we present here the following definitions. All symbols involved in these developments are depicted in Table 2.

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,

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

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

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

where $\mathbf{p}$, $\mathbf{v}$, $\mathbf{q}$ are position, velocity and quaternion orientation (refer [13] 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 $\delta\theta \in \mathfrak{so}3 \subset \mathbb{R}^3$ 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,

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

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

where $\delta\mathbf{q} = \mathbf{q}\{\delta\theta\} \triangleq \exp(\delta\theta/2)$ is the orientation error in $SO(3)$ expressed as a unit quaternion —see [16] for details.

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 3 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).

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

Each sensor, used in the proposed estimation pipeline, presents a different measurement type and statistical characteristics. For the sake of simplicity we refer the reader to [16] for detailed descriptions of the observations models and Jacobians. These observation models are used in the filter propagation (IMU) and correction (smart camera and range sensor) steps. All filter steps are also descibed in [16].

Our platform of choice is a quadrotor due to its mechanical simplicity and ease of control. We 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 $SE(3)$ [5].

4 Simulations

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 [2] 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[3]. The optimized high-rate C++ implementation is available upon request.

Table 3 shows both position RMSE and the abovementioned orientation error metric (no units) achieved at the end of $N = 20$ simulated flights of almost

[2] `https://gitlab.iri.upc.edu/asantamaria/QuadSim`

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

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 3 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 re-normalization 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.

5 Experiments

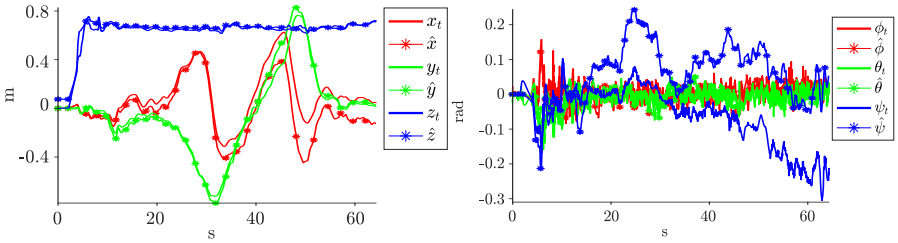
The quadrotor used in the real experiments is the ASCTEC Hummingbird research platform shown in Fig. 1. This platform has an off-the-shelf built-in IMU running at 100 Hz, and we equipped it with a PX4-Flow smart camera [3], with a rate of 200 Hz, and a TeraRangeOne range sensor [12] with a frequency of up to 800 Hz.

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[4] 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. 3(a) and 3(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. 3(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. Finally, 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.

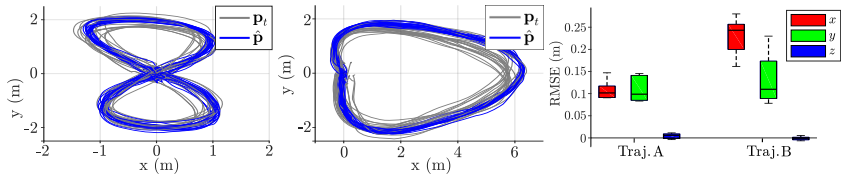
Fig. 4 shows experiments for two different trajectories, 4(a) and 4(b). We launched 25 autonomous runs for each trajectory with a desired height of 1 m and

[4] `www.qualisys.com`



(a) Position (b) Orientation

Fig. 3 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].



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

Fig. 4 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).

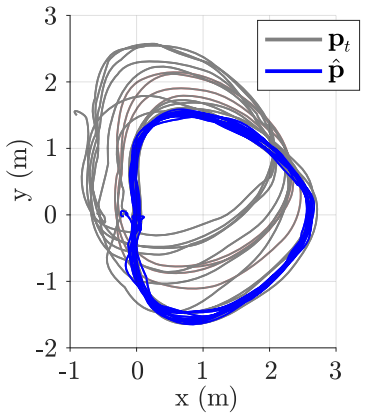


Fig. 5 Position estimation results for a long experiment (almost 10 min of continuous flight and a full battery discharge). Note that in full autonomous mode the vehicle is controlled using the estimation, thus the drift of the platform can be seen in the ground-truth trajectory (Qualisys motion capture system).

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. 4(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 method 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. 5 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. 5) is used to control the vehicle, thus the estimation errors are reflected in the plot of the ground-truth trajectory (gray in Fig. 5). 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.

6 Conclusions

In this chapter, we presented a state estimator design for aerial manipulators 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 and show the feasibility of using such low-cost sensor setup with light algorithms to achieve not only hovering maneuvers but also fully autonomous navigation. Our vast experimentation allows us to conclude that the effects of all the variations in the estimator design are minimal. In particular, the refinements on the transition matrices $\mathbf{F}_1, \mathbf{F}_2, \mathbf{F}_3$, 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 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 F1, Q0B and quaternion re-normalization; or the more proper ESKF with F1, 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

aerial manipulator to perform a number of tasks that require navigation without the aid on any external positioning system.

References

1. M. Blösch, S. Weiss, D. Scaramuzza, and R. Siegwart. Vision based mav navigation in unknown and unstructured environments. In *2010 IEEE International Conference on Robotics and Automation*, pages 21–28, May 2010.
2. Joel A Hesch, Dimitrios G Kottas, Sean L Bowman, and Stergios I Roumeliotis. Camera-imu-based localization: Observability analysis and consistency improvement. *The International Journal of Robotics Research*, 33(1):182–201, 2014.
3. D. Honegger, L. Meier, P. Tanskanen, and M. Pollefeys. An open source and open hardware embedded metric optical flow cmos camera for indoor and outdoor applications. In *2013 IEEE International Conference on Robotics and Automation*, pages 1736–1741, May 2013.
4. Jonathan Kelly and Gaurav S. Sukhatme. Visual-inertial sensor fusion: Localization, mapping and sensor-to-sensor self-calibration. *International Journal of Robotics Research*, 30(1):56–79, Jan 2011.
5. T. Lee, M. Leok, and N. H. McClamroch. Nonlinear robust tracking control of a quadrotor uav on se(3). In *2012 American Control Conference (ACC)*, pages 4649–4654, June 2012.
6. M. Li and A. I. Mourikis. Improving the accuracy of ekf-based visual-inertial odometry. In *2012 IEEE International Conference on Robotics and Automation*, pages 828–835, May 2012.
7. Mingyang Li and Anastasios I. Mourikis. High-precision, consistent ekf-based visual-inertial odometry. *The International Journal of Robotics Research*, 32(6):690–711, 2013.
8. Venkatesh K Madyastha, Vishal C Ravindra, Srinath Mallikarjunan, and Anup Goyal. Extended Kalman filter vs. error state Kalman filter for aircraft attitude estimation. In *AIAA Guidance, Navigation and Control Conference*, pages 6615–6638, Portland, August 2011.
9. Vishal C. Ravindra, Venkatesh K. Madyastha, and Anup Goyal. The equivalence between two well-known variants of the Kalman filter. In *International Conference on Advances in Control and Optimization of Dynamic Systems*, Feb 2012.
10. R. Rossi, A. Santamaria-Navarro, J. Andrade-Cetto, and P. Rocco. Trajectory generation for unmanned aerial manipulators through quadratic programming. *IEEE Robotics and Autom. Letters*, 2(2):389–396, April 2017.
11. Cyril Roussillon, Aurélien Gonzalez, Joan Solà, Jean Marie Codol, Nicolas Mansard, Simon Lacroix, and Michel Devy. RT-SLAM: A generic and real-time visual SLAM implementation. In *Computer Vision Systems*, volume 6962 of *Lect. Notes in Comp. Science*, pages 31–40. Springer Berlin Heidelberg, 2011.
12. M Ruffo, M Di Castro, L Molinari, R Losito, A Masi, J Kovermann, and Luis Rodrigues. New infrared time-of-flight measurement sensor for robotic platforms. In *International Symposium and International Workshop on ADC Modelling and Testing*, pages 13–18, September 2014.
13. A. Santamaria-Navarro, P. Grosch, V. Lippiello, J. Sol, and J. Andrade-Cetto. Uncalibrated visual servo for unmanned aerial manipulation. *IEEE/ASME Transactions on Mechatronics*, 22(4):1610–1621, Aug 2017.
14. A. Santamaria-Navarro, V. Lippiello, and J. Andrade-Cetto. Task priority control for aerial manipulation. In *IEEE International Symposium on Safety, Security, and Rescue Robotics (SSRR)*, pages 1–6, Oct 2014.
15. A. Santamaria-Navarro, J. Solà, and J. Andrade-Cetto. High-frequency mav state estimation using low-cost inertial and optical flow measurement units. In *Intelligent Robots and Systems (IROS), 2015 IEEE/RSJ International Conference on*, pages 1864–1871, Sept 2015.
16. Angel Santamaria-Navarro, Giuseppe Loianno, Joan Solà, Vijay Kumar, and Juan Andrade-Cetto. Autonomous navigation of micro aerial vehicles using high-rate and low-cost sensors. *Autonomous Robots, To appear.*, Dec 2018.

17. Nikolas Trawny and Stergios I. Roumeliotis. Indirect Kalman filter for 3D attitude estimation. *University of Minnesota, Dept. of Computer Science & Engineering, Technical Report*, 2, rev. 57, 2005.
18. Stephan Weiss, Davide Scaramuzza, and Roland Siegwart. Monocular-SLAM-based navigation for autonomous micro helicopters in gps denied environments. *Journal of Field Robotics*, 28(6):854–874, 2011.