Estimation, Control, and Planning for Aggressive Flight With a Small Quadrotor With a Single Camera and IMU

Giuseppe Loianno, Chris Brunner, Gary McGrath, and Vijay Kumar

***Abstract*—We address the state estimation, control, and planning for aggressive flight with a 150 cm diameter, 250 g quadrotor equipped only with a single camera and an inertial measurement unit (IMU). The use of smartphone grade hardware and the small scale provides an inexpensive and practical solution for autonomous flight in indoor environments. The key contributions of this paper are: 1) robust state estimation and control using only a monocular camera and an IMU at speeds of 4.5 m/s, accelerations of over 1.5 g, roll and pitch angles of up to 90°, and angular rate of up to 800°/s without requiring any structure in the environment; 2) planning of dynamically feasible three-dimensional trajectories for slalom paths and flights through narrow windows; and 3) extensive experimental results showing aggressive flights through and around obstacles with large rotation angular excursions and accelerations.**

***Index Terms*—Aerial robotics, optimization and optimal control, sensor-based control.**

I. Introduction

MICRO Aerial Vehicles (MAVs) equipped with on-board sensors, are ideal platforms for autonomous navigation in complex and confined environments for solving tasks such as exploration [1], inspection [2], [3], mapping [4], interaction with the environment [5], [6] and, search and rescue [7].

In this work, we analyze the problem of aggressive maneuvers with a small and lightweight quadrotor platform using on-board sensing capabilites such as a single camera and IMU. Aggressive and fast maneuvers have always been considered an active research area from a control and estimation perspectives due to the extreme flight conditions, an underactuated system as a quadrotor, is subjected to. The ability to fly fast and maneuver aggressively is useful in tasks with time constraints and in search and rescue applications. A number of groups have demonstrated aggressive maneuvers with aerial vehicles such as quadrotors [8]–[10]. These works mostly show strategies for generating sequences of controllers that stabilize the robot to a desired state. In [9], the authors focus on the hovering stabilization problem after a flip maneuver, whereas [8], [10] address the learning and adaptive iteration of the control law, which is refined based on different flight trials. Excellent results, showing aggressive maeuvers with a quadrotor have been obtained in [11]. Compared to previous works, the main difference is in the use of the property of differential flatness and the development of an efficient planning algorithm to generate trajectories for a quadrotor without the limitation to first order dynamics. The aggressive motion task is solved via the composition of three control modes. The main drawbacks of this approach are the use of a motion capture systems for localization and linearized controllers.

In this work, we solve the aggressive maneuvering task using only on-board sensing capabilities. The minimal sensor suite for autonomous localization consists of two inexpensive, lightweight and widely available sensors, a monocular camera and an IMU as shown in [12]–[15]. The observability analysis applied to aerial navigation is discussed in [16], [17]. Relevant to this work, are also results focusing on vision-based fast navigation with MAVs. In [18], a quadrotor equipped with a stereo camera and IMU is able to fly at 4 m/s, whereas in [19], a smartphone is able to provide the necessary computation capabilities to fly up to 3 m/s using the single camera and IMU available on the device. The setup in [18] leverages a stereo camera config-

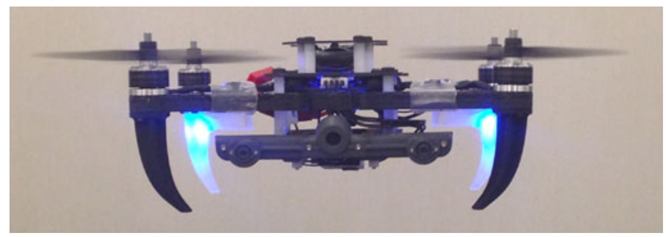

Fig. 1. A 250 gram quadrotor equipped with a single downward-facing camera, a forward-facing stereo camera (not used in this paper), an IMU, and a Qualcomm Snapdragon^{TM} and Hexagon^{TM} DSP.

Manuscript received September 10, 2016; accepted November 5, 2016. Date of publication November 29, 2016; date of current version December 26, 2012. This paper was recommended for publication by Associate Editor P. Pounds and Editor J. Roberts upon evaluation of the reviewers' comments. This work was supported in part by the Qualcomm Research, in part by the ARL Grant W911NF-08-2-0004, in part by the ONR Grants N00014-07-1-0829 and N00014-14-1-0510, in part by the ARO Grant W911NF-13-1-0350, in part by the NSF Grants IIS-1426840 and IIS-1138847, in part by the DARPA Grants HR001151626 and HR0011516850, and in part by the TerraSwarm, one of six centers of STARnet, a Semiconductor Research Corporation program sponsored by MARCO and DARPA.

G. Loianno and V. Kumar are with the General Robotics, Automation, Sensing, and Perception Laboratory, University of Pennsylvania, Phialdelphia, PA 19103 USA (e-mail: loiannog@seas.upenn.edu; kumar@seas.upenn.edu).

C. Brunner and G. McGrath are with Qualcomm Technologies, Inc., San Diego, CA 92121 USA (e-mail: chris@qti.qualcomm.com; gmcgrath@qti.qualcomm.com).

This paper has supplementary downloadable material available at http://ieeexplore.ieee.org, provided by the authors.

Color versions of one or more of the figures in this letter are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/LRA.2016.2633290

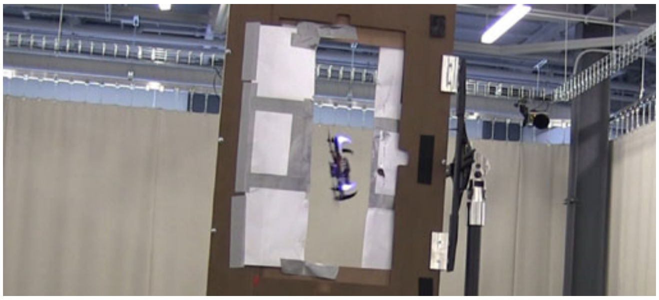

Fig. 2. The platform traversing a vertical narrow window gap.

uration, which facilitates the handling of the scaling problems that arise with monocular camera systems. In [20] the authors focus on the 6-DOF pose tracking with a DVS camera. This method works only with planar shapes or gradient maps that are known a priori. In addition, the estimated pose is not used for closed loop control. Instead an external motion capture system is used. Finally, in [21] an approach for automatically recover from any unknown initial attitude is proposed. However, the system employs a downward pointing laser for scale estimation and only focuses on the specific maneuver for recovery.

In this paper, we address the perception, control and planning problems to enable autonomous aggressive maneuver with small scale, low power and lightweight quadrotor with limited computational capabilities. Our platform of choice is a 0.15 m diameter, 250 g quadrotor using Qualcomm Snapdragon™Flight™ (see Fig. 1). To validate the proposed strategy, we focus on aggressive flights with large operating envelope and with significant excursions in roll and pitch angles such as flights through narrow window gaps with different orientations as shown in Fig. 2. Compared to previous approaches, our solution relies on a nonlinear controller and on on-board capabilities without requiring the need of external motion capture systems. Moreover, we do not rely on switching and learning strategies, but we propose a deterministic trajectory planning approach to generate aggressive maneuvers with a small quadrotor. These motions require fast and large rotations and accelerations. Finally, in all previous works medium size quadrotor, weighing more than 500 g, are used. We believe this is the smallest, fully autonomous flying robot in which all sensing and computing is done on-board without GPS.

This work presents multiple contributions. First, we develop the control, estimation pipelines to enable vision-based aggressive maneuvers with a single camera and IMU. Second, we show how to generate dynamically feasible trajectories enabling flights through and around obstacles in constrained trajectories without the need to switch between different control modes. Finally, this is the first time that perception, planning and control are combined for autonomous navigation and aggressive maneuvers of a small lightweight aerial vehicle without relying on GPS and on any external motion capture system, but just using on-board computation. The proposed operating conditions require perception, state estimation, environment reconstruction, obstacle avoidance and trajectory planning algorithms with large accelerations and rotations over short ranges and time scales. Thus, the perception, planning and control subproblems have to be solved concurrently as a single problem. Our solution enables vision-based closed loop control flights with roll and pitch angular values up to 90° without using surfaces with special texture.

The paper is organized as follows. Section II shows a general overview of our framework. In Section III, the dynamics of the quadrotor and the control framework are provided, whereas in Section IV, the strategy to obtain the pose of the vehicle at high rate, enabling autonomous flight, is shown. Section V, presents our methodology to generate dynamically feasible trajectories through and around obstacles in constrained trajectories. Section VI presents extensive results on navigation with the proposed prototype. Section VII concludes the work and provides an overview of the multiple future scenarios.

II. System Overview

Our platform of choice is a quadrotor due to its mechanical simplicity [22] due to its ability to operate in confined spaces, hover in place and perch or land on flat surfaces.

A. Hardware Architecture

The experimental platform shown in Fig. 2 is equipped with 4 brushless motors and a Qualcomm Snapdragon™ board. This board features a Qualcomm Hexagon™ DSP, Wi-Fi, Bluetooth connectivity, 802.11n Wi-Fi, and GPS, all packed into a board (58 mm × 40 mm). Based on the Snapdragon™ 801 processor, the system just uses 1 core of the total CPU. The board is equipped with a downward facing VGA camera with 160° field of view, a VGA stereo camera, and a 4 K camera.

B. Software Architecture

The software components are a position and attitude controller (see Fig. 3), a state estimation algorithm composed of VIO described in Section IV, which processes images at 30 Hz and an Unscented Kalman Filter (UKF) to deal with control constraints for fast motions (see green box in Fig. 3). The control receives the estimated pose at 500 Hz from the UKF and sends the attitude commands to the DSP. It is worth to specify that the presented approach, for state estimation employs only the downward facing camera and the IMU available on the Qualcomm Snapdragon™Flight™. The framework has been developed in ROS.[1] All the tasks are then executed on-board the vehicle in separate threads with state estimation and control at a fixed rate of 500 Hz.

III. Modeling and Control

We introduce the system dynamics and the control employed to execute aggressive maneuvers.

A. Dynamic Model

Consider an inertial reference frame denoted by $\begin{bmatrix} \boldsymbol{e}_1, \boldsymbol{e}_2, \boldsymbol{e}_3 \end{bmatrix}$ and a body reference frame centered in the center of mass of the vehicle denoted by $\boldsymbol{R} = \begin{bmatrix} \boldsymbol{b}_1, \boldsymbol{b}_2, \boldsymbol{b}_3 \end{bmatrix}$ (see Fig. 4) where $\boldsymbol{R} \in SO(3)$. The system dynamic model is

$$\begin{aligned} \dot{\boldsymbol{x}} &= \boldsymbol{v}, m\dot{\boldsymbol{v}} = \boldsymbol{R}\tau\boldsymbol{e}_3 - mg\boldsymbol{e}_3, \\ \dot{\boldsymbol{R}} &= \boldsymbol{R}\hat{\boldsymbol{\Omega}}, \boldsymbol{J}\dot{\boldsymbol{\Omega}} + \boldsymbol{\Omega} \times \boldsymbol{J}\boldsymbol{\Omega} = \boldsymbol{M}, \end{aligned} \tag{1}$$

[1] www.ros.org

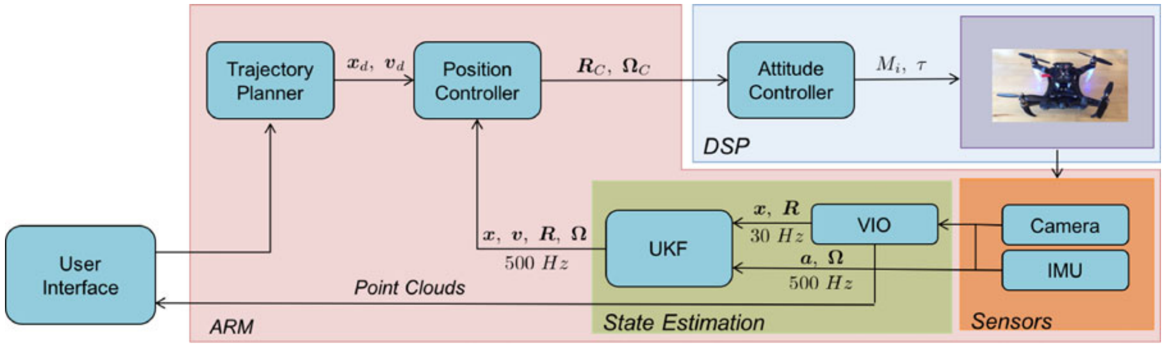

Fig. 3. The architecture overview.

Fig. 4. The quadrotor model and its reference frames.

where $\boldsymbol{x} \in \mathbb{R}^3$ is the Cartesian position of the vehicle expressed in the inertial frame, $\boldsymbol{v} \in \mathbb{R}^3$ is the velocity of the vehicle in the inertial frame, $m \in \mathbb{R}$ is the mass, $\boldsymbol{\Omega} \in \mathbb{R}^3$ is the angular velocity and $\boldsymbol{J} \in \boldsymbol{R}^{3\times3}$ is the inertia matrix both with respect to the body frame. The hat symbol $\hat{\cdot}$ denotes the skew-symmetry operator according to $\hat{\boldsymbol{x}}\boldsymbol{y} = \boldsymbol{x} \times \boldsymbol{y}$ for all $\boldsymbol{x}, \boldsymbol{y} \in \mathbb{R}^3$, g is the standard gravitational acceleration.

B. Position and Attitude Controllers

In most previous works, a back-stepping approach is used for control because the attitude dynamics can be assumed to be faster than the position dynamics, and linearized controllers are used for both loops [16], [22]. In this work, because we need to model aggressive maneuvers and large excursions from the hover position, we use a nonlinear controller based on [23], [24].

The control inputs τ, M are chosen as

$$\begin{aligned} \boldsymbol{M} =& -k_R \boldsymbol{e}_R - k_\Omega \boldsymbol{e}_\Omega + \boldsymbol{\Omega} \times J\boldsymbol{\Omega} \\ & - \boldsymbol{J}\left(\hat{\boldsymbol{\Omega}}\boldsymbol{R}^\top \boldsymbol{R}_C \boldsymbol{\Omega}_C - \boldsymbol{R}^\top \boldsymbol{R}_C \dot{\boldsymbol{\Omega}}_C\right), \\ \tau =& \left(-k_x \boldsymbol{e}_x - k_v \boldsymbol{e}_v + mg\boldsymbol{e}_3 + m\ddot{\boldsymbol{x}}_d\right) \cdot \boldsymbol{R}\boldsymbol{e}_3, \end{aligned} \tag{2}$$

with $\ddot{\boldsymbol{x}}_d$ the desired acceleration, k_x, k_v, k_R, k_Ω positive definite terms. The subscript C denotes a commanded value. The relationship between single motor force f_j, the total thrust τ and the total moment $\boldsymbol{M}$ can be inverted for non-zero values of the distance from the center of mass to the center of each rotor. Our assumption that τ and $\boldsymbol{M}$ are the inputs of the plant is therefore valid. The quantities $\boldsymbol{e}_R, \boldsymbol{e}_\Omega, \boldsymbol{e}_x, \boldsymbol{e}_v$ are the orientation, angular rate, and translation errors respectively, defined in [19], [23], [24]. If the initial attitude error is between 90° and 180°, the zero equilibrium of the tracking errors is almost globally exponentially attractive [23].

IV. State Estimation

In this section, we describe the different steps that enable to recover the 6-DOF (Degree Of Freedom) pose of the vehicle, necessary to control the vehicle. Our pipeline uses an EKF combining visual and IMU data (VIO block in Fig. 3). In this work, to enable aggressive maneuvers, we need a high level of robustness and high control rates. In addition to the EKF, an Unscented Kalman Filter (UKF) is able to estimate the full state of the vehicle at 500 Hz. We use an UKF, instead of an EKF because of the need to operate over a large envelope with significant excursions in roll and pitch angles from the hover configuration and velocities up to 5 m/s. The separation of the VIO and UKF is useful to keep the CPU usage limited. The state size of the VIO algorithm is not constant, since image features are part of it. For this reason, considering a prediction at 500 Hz is more expensive than 30 Hz. In this way, we obtain similar performances satisfying at the same time the control rate constraints. The state estimation rate (over 100 Hz) also guarantees small integration errors and thus better accuracy.

A. Visual Inertial Odometry

The VIO system localizes the rigid body with respect to the inertial frame using accelerometers, gyroscopes and camera sensors. The navigation state vector $\mathbf{x}(t) \in \mathbb{R}^{12} \times \boldsymbol{se}(\mathbf{3})$ is

$$\mathbf{x} = \left[\boldsymbol{x}^\top\ \boldsymbol{\Theta}^\top\ \boldsymbol{v}^\top\ \boldsymbol{\gamma}^\top\ \boldsymbol{b}_g^\top\ \boldsymbol{b}_a^\top\right]^\top, \tag{3}$$

where $\boldsymbol{x} \in \mathbb{R}^3$, $\boldsymbol{v}$ denote the robot's position and velocity respectively as expressed in Section III-A, The term $\boldsymbol{\Theta}$ is the attitude $\boldsymbol{R}$ expressed in exponential coordinates, $\boldsymbol{\gamma}$ is the unknown gravity vector in the inertial frame, and $\boldsymbol{b}_a$ and $\boldsymbol{b}_g$ denote accelerometer and gyroscope biases modeled as random walk processes. The prediction step is based on the IMU integration. The measurement update step is given by the standard 3D landmark perspective projection onto the image plane leading to the EKF updates, assuming that distinguishable features that can be tracked over time using a camera. In the past decade, camera measurements have been effectively used to aid Inertial Navigation Systems [12]–[14]. The setup also estimates the calibration parameters, such as inertial sensor scale factor, non-orthogonality, camera-accelerometer transformation by appending them to the state. We also employ an inverse depth parametrization [25], which facilitates feature initialization and the convergence of the corresponding 3D point. If the stationary feature is found to be persistent, then we augment the state with the feature vector along with the pose of the body frame at which it was first observed in order to correctly account for correlations of errors in subsequent measurements of the feature to errors in the state.

B. Unscented Kalman Filter

To enable on–board control, an UKF is used to estimate the full *state* of the vehicle at 500 Hz. The state is represented by

$$\mathbf{x}_f = \begin{bmatrix} \boldsymbol{x}^\top & \boldsymbol{v}^\top & \Phi^\top & \boldsymbol{b}_a^\top \end{bmatrix}^\top, \tag{4}$$

where $\boldsymbol{x}$, $\boldsymbol{v}$ have been defined in Section III-A, Φ is the quaternion and $\mathbf{b}_a$ the accelerometer biases. The prediction step uses the input $\boldsymbol{y}_a, \boldsymbol{y}_g \in \mathbb{R}^3$ linear acceleration and angular velocity measurements given by the IMU. The VIO pose estimates, are then used to update the state estimate using a linear measurement model. The measurement delay due to the image processing is compensated accumulating the IMU values till a new measurement from the VIO block is provided.

V. Planning for Aggressive Maneuvers

The trajectory planning has to generate dynamically feasible trajectories that will take the quadrotor to its destination. In this section, we focus on two types of aggressive maneuvers.

A. Dynamically Feasible Trajectories

Similar to [24], we use the differential flatness property, which facilitates the computation of trajectories for the underactuated quadrotor system. We propose the a set of variables called flat outputs, the Cartesian position vector $\boldsymbol{x}$ and the yaw angle ψ, which will be used to show that the system's state and the control inputs can be written in terms of this subset of variables and their derivatives. Considering (1), the nominal force is

$$\tau = m \left\| \ddot{\boldsymbol{x}} + g\boldsymbol{e}_3 \right\|, \tag{5}$$

and the orientation of the third axis of body frame, $\boldsymbol{b}_3$ is

$$\boldsymbol{b}_3 = \frac{\ddot{\boldsymbol{x}} + g\boldsymbol{e}_3}{\left\| \ddot{\boldsymbol{x}} + g\boldsymbol{e}_3 \right\|}. \tag{6}$$

The rest of the rotation matrix, $\boldsymbol{R}$, can be determined by defining a vector $\boldsymbol{b}_c$, using the yaw angle ψ, to determine $\boldsymbol{b}_2$

$$\boldsymbol{b}_2 = \frac{\boldsymbol{b}_3 \times \boldsymbol{b}_c}{\left\| \boldsymbol{b}_3 \times \boldsymbol{b}_c \right\|}, \boldsymbol{b}_c = \begin{bmatrix} \cos\psi & \sin\psi & 0 \end{bmatrix}. \tag{7}$$

Then $\boldsymbol{b}_2 \times \boldsymbol{b}_3$ gives $\boldsymbol{b}_1$. Differentiating again (1) we obtain

$$m\boldsymbol{x}^{(3)} = -\tau \dot{\boldsymbol{R}} \boldsymbol{e}_3 - \dot{\tau} \boldsymbol{R} \boldsymbol{e}_3 = -\tau \boldsymbol{R} \hat{\boldsymbol{\Omega}} \boldsymbol{e}_3 - \dot{\tau} \boldsymbol{b}_3 \tag{8}$$

and the scalar projection onto $\boldsymbol{b}_3$ reveals that

$$\dot{\tau} = \boldsymbol{b}_3 \cdot m\boldsymbol{x}^{(3)}. \tag{9}$$

Next, we can determine the first two terms of the angular body rates $\boldsymbol{\Omega}$, by solving (8) for $\hat{\boldsymbol{\Omega}}\boldsymbol{e}_3$

$$\begin{bmatrix} \Omega_1 \\ \Omega_2 \end{bmatrix} = \frac{m}{\tau} \begin{bmatrix} -\boldsymbol{b}_2^\top \\ \boldsymbol{b}_1^\top \end{bmatrix} \boldsymbol{x}^{(3)}. \tag{10}$$

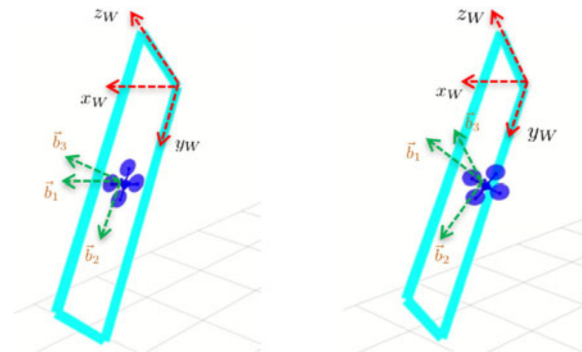

Fig. 5. The planned trajectory over the window with 45° roll and 20° pitch angles, planning (left) and without (right) planning the yaw angle.

The third term of $\boldsymbol{\Omega}$ depends on the yaw angle derivative $\dot{\psi}$. Decomposing $\boldsymbol{R}$ by the Euler angles $Z(\psi) - X(\phi) - Y(\theta)$ parametrization we obtain

$$\Omega_3 = \Omega_1 \tan\theta + \frac{\cos\phi}{\cos\theta}\dot{\psi} \tag{11}$$

Differentiating again (8) and similar to before, we can solve for the first two component of $\dot{\boldsymbol{\Omega}}$. The third element will require the 2nd derivative of the yaw angle. With this angular acceleration, we can solve for the required moments. Thus, the control inputs can be computed in terms of the flat outputs and their derivatives. The 4^{th} derivative of position appears in the control inputs and the 2^{nd} derivative of the yaw appears in the moments. An optimal motion plan is defined as one that that minimizes the cost function Γ

$$\Gamma = \int_{t_0}^{t_f} \left\| \frac{d^4 \boldsymbol{x}(t)}{dt^4} \right\|^2 dt + \left\| \frac{d^2 \psi(t)}{dt^2} \right\|^2 dt. \tag{12}$$

Considering a n^{th} order time-parametrized polynomial trajectories for each Cartesian coordinate d and the yaw

$$\alpha_d(t) = \sum_{i=0}^{n} c_{i_d} t^i = \boldsymbol{c}_d^\top \boldsymbol{t},\ d = 1, \cdots, 4, \tag{13}$$

the problem in eq. (12) can be formulated as a Quadratic Programming (QP), with the initial t_0 and the final t_f times

$$\min \boldsymbol{c}_d^\top \left(\int_{t_0}^{t_f} \frac{d^4}{dt^4}\boldsymbol{t} \left(\frac{d^4}{dt^4}\boldsymbol{t} \right)^\top dt \right) \boldsymbol{c}_d = \min \boldsymbol{c}_d^\top \mathcal{T}_d \boldsymbol{c}_d \tag{14}$$

and in the general form

$$\begin{aligned} &\min C^\top \mathcal{T} C, \\ &\text{subject to } AC \leqslant B, A_e C = B_e, \end{aligned} \tag{15}$$

with

$$C = \begin{bmatrix} \boldsymbol{c}_1^\top & \boldsymbol{c}_2^\top & \boldsymbol{c}_3^\top & \boldsymbol{c}_4^\top \end{bmatrix}^\top, \mathcal{T} = \text{diag}\begin{bmatrix} \mathcal{T}_1 & \mathcal{T}_2 & \mathcal{T}_3 & \mathcal{T}_4 \end{bmatrix}.$$

The matrix $A \in \mathbb{R}^{k \times 4n}$ and $B \in \mathbb{R}^k$, where k is the total number of linear constraints. The matrix $A_e \in \mathbb{R}^{p \times 4n}$ and vector $B_e \in \mathbb{R}^p$ can be used to impose p equality constraints. Generally, a trajectory is divided into segments. The coefficients for each segment can be easily incorporated into the QP. The equality constraints are used to guarantee during the trajectory the

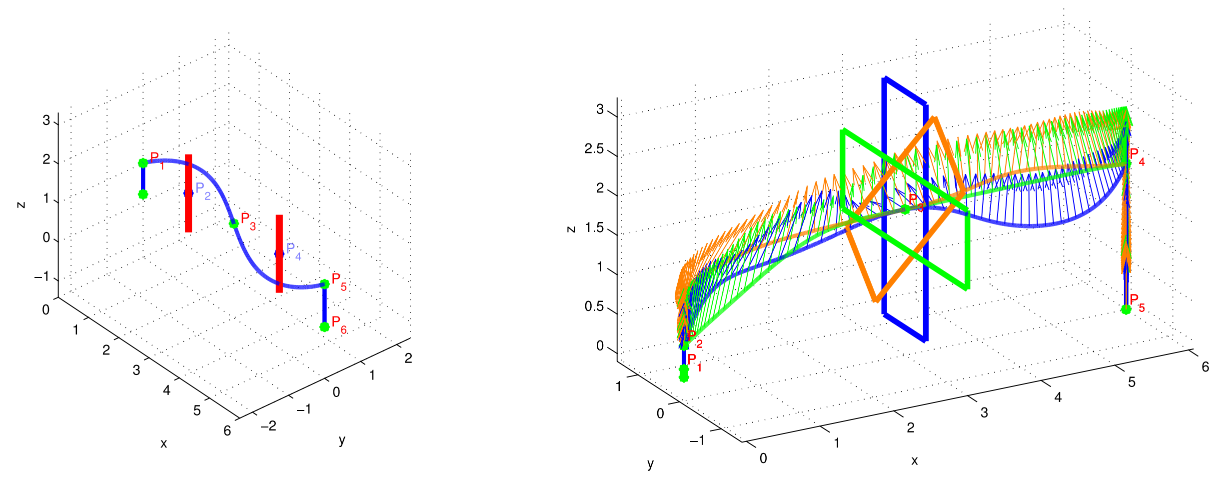

Fig. 6. The slalom path with two poles obstacles in red (left) and the planned trajectory over the narrow window gap at 0° green − 45° orange − 90° blue (right) with the arrow indicating the desired acceleration direction in 3D Cartesian space.

continuity of position and its derivatives and to specify a constraint at a desired time. We also consider a maximum velocity bound $\boldsymbol{v}_{\max}$.

It is also necessary to consider constraints imposed by the mechanics and sensors of the system since we are focusing on aggressive maneuvers. Specifically, the maximum thrust imposes a constraint of the type

$$m\|\ddot{\boldsymbol{x}} + g\boldsymbol{e}_3\| \leqslant \tau_{\max}, \forall t \in [t_0, t_f], \tag{16}$$

where $\tau_{\max} = \sum_{j=1}^{4} f_{j_{\max}} = 4k_f\omega_{m_{\max}}^2$ is the total thrust acting in the $\boldsymbol{b}_3$ direction, whereas k_f is the thrust coefficient and $\omega_{m_{\max}}$ the maximum single motor speed.

In addition, a constraint on the jerk is induced by the maximum angular speed detectable by the gyros as

$$f_1\left(\boldsymbol{x}^{(3)}, \ddot{\boldsymbol{x}}, \dot{\psi}\right) \leqslant \omega_{\max}, \forall t \in [t_0, t_f], \tag{17}$$

and the maximum moment $M_{\max}$, on one body axis, induces a constraint as

$$f_2\left(\boldsymbol{x}^{(4)}, \boldsymbol{x}^{(3)}, \ddot{\boldsymbol{x}}, \dot{\psi}, \ddot{\psi}\right) \leqslant M_{\max}, \forall t \in [t_0, t_f], \tag{18}$$

where $M_{\max} = l\left(\omega_{m_{\max}} - \omega_{m_{\min}}\right)$, with l the vehicle arm length. Practically, the constraints expressed by (16), (17), (18) are nonlinear, thus they cannot be directly incorporated in the QP problem. However, to take advantage of the QP formulation, it is still possible to consider them indirectly with upper bounds on the acceleration, jerk and snap respectively and then verifying *a-posteriori* they are respected.

B. Slalom Path

We consider maneuvers with increasing degree of difficulties in term of acceleration, velocity and attitude angles. In this case, we assume to know the obstacles' locations with respect to the robot's starting position. We consider the starting point P_1 and the final point P_5. A set of obstacles is located between these two points (in our case 2 static obstacles are located between $P_1 - P_3$ and $P_3 - P_5$ see Fig. 6 left). The obstacle's position imposes some constraints on the planned trajectory. We consider a set of virtual waypoints $P_{v_i}\left(P_{v_{i_x}}, P_{v_{i_y}}, P_{v_{i_z}}\right)$ located at the obstacle positions (P_2 and P_4 in the considered case) and their corresponding times instants t_{v_i}. In addition to the constraints specified in the previous paragraph, we can then impose in our QP problem a bound on each of the Cartesian dimension (x, y, z) around the obstacles' positions as

$$x(t) \geqslant P_{v_{i_x}} + \epsilon_x, \quad y(t) \geqslant P_{v_{i_y}} + \epsilon_y,\ z(t) \geqslant P_{v_{i_z}} + \epsilon_z,$$
$$\forall t \in \left(t_{v_i} - \delta, t_{v_i} + \delta\right), \tag{19}$$

where δ is a time interval and ϵ_i, ϵ_y, ϵ_z indicate the thresholds the vehicle's center of mass should respect to be safe.

C. Planning Trajectories through a Narrow Gap

Let us consider a narrow obstacle like a window. We assume the window gap is in the field of view at the robot's starting position. The stereo configuration available on the board allows to identify the window using a planar homography algorithm. This gives as output the rotation $\boldsymbol{R}_W^B = [\boldsymbol{x}_W\ \boldsymbol{y}_W\ \boldsymbol{z}_W]$ and the translation $\boldsymbol{t}_W$ of the window with respect to the initial robot frame B. Without loss of generality let us decompose the rotation using the $Z(\psi) - X(\phi) - Y(\theta)$ Euler angles parametrization. It is worth specifying that our approach does not depend on the Euler parametrization since it is based on a geometric control approach as shown in Section III. However, it is necessary to employ a parametrization in case a yaw angle has to be planned. The parametrization of choice should then be the same used for the differential flatness in Section V-A. It is useful to clarify again that neither the stereo camera nor the shape of the window is used to solve for visual scale. Since the gap is narrow, the vehicle can only traverse it such that the main body plane (the plane defined by the centers of the four propellers) is orthogonal to the window planar surface defined by the $\boldsymbol{y}_W - \boldsymbol{z}_W$ axes (see Fig. 5). However, the user can decide if the yaw should be zero at the traversing point. In that case, the vehicle will traverse the window with the x body axis in the same direction of the window axis (see Fig. 5 left), otherwise the vehicle will have to keep constant yaw during the trajectory and it should then use both roll and pitch to traverse the window (see Fig. 5 right) such to respect that the body plane is orthogonal to the $\boldsymbol{y}_W - \boldsymbol{z}_W$ plane. Since we are planning already aggressive trajectories we opt for the fist solution and solve the yaw offset

TABLE I
POSITION AND VELOCITY RMSE AND STD OF THE ESTIMATES COMPARED TO MOTION CAPTURE SYSTEM AT FOR THE SLALOM CASE

Obstacles apart (m)	Max. Acc. (m/s^2)	Max. Ang. Vel. (deg/s)	Cartesian Component	RMSE VIO estimation (m)	RMSE UKF estimation (m)	RMSE Velocity (m/s)	STD position VIO (m)	STD position UKF (m)	STD velocity (m/s)
3	4.11	240.0653	x	0.0774	0.0793	0.0868	0.0758	0.0645	0.0851
			y	0.0290	0.0436	0.0813	0.0228	0.0812	0.0700
			z	0.0662	0.0635	0.0916	0.0290	0.0915	0.0641
1.5	5.69	344.8861	x	0.0437	0.0490	0.0421	0.0423	0.0391	0.0355
			y	0.0388	0.0374	0.0625	0.0325	0.0281	0.0424
			z	0.0637	0.0558	0.0538	0.0438	0.0348	0.0413

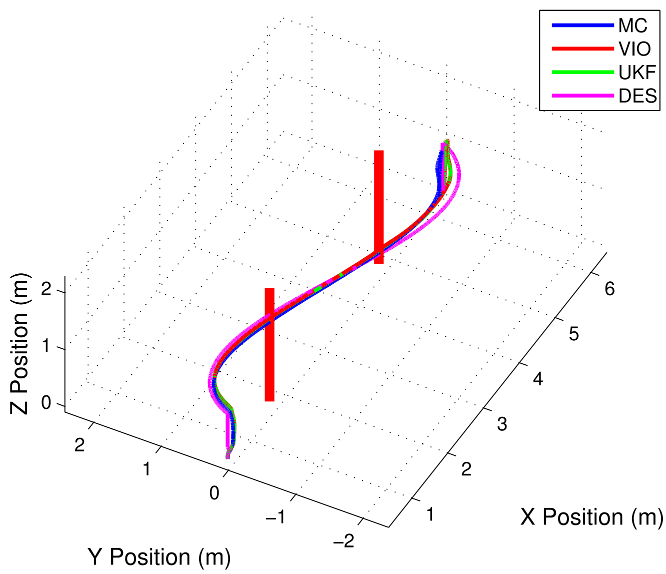

Fig. 7. Ground truth (blue), estimated (green), desired (magenta) trajectories for the 6 m slalom trajectory

between the window and the vehicle during the takeoff phase. All the information to plan the trajectory through the window is then available. We define the traversing window point as $\boldsymbol{t}_W$ and we enforce the orientation $\boldsymbol{R}_W^B$ at the gap with the following acceleration constraint

$$\boldsymbol{a}_{W_{\mathrm{des}}} = {\boldsymbol{R}_W^B}^\top \boldsymbol{a}_W - \gamma \boldsymbol{e}_3, \tag{20}$$

where the $\boldsymbol{a}_W = a\boldsymbol{e}_3$ is chosen by the user according to what the desired vertical acceleration is. The trajectory is planned through points $P_1 - P_5$ (see Fig. 6) with P_1 and P_5 the starting and landing point respectively. This constraint is derived from (6). The third body axis is acceleration and gravity vectors dependent. This guarantees to plan $\boldsymbol{b}_3$ as in (6) and also $\boldsymbol{b}_2$ as in (7). Then, $\boldsymbol{b}_1$ is chosen as $\boldsymbol{b}_2 \times \boldsymbol{b}_3$. We estimate that an orientation time change by $\pi/2$ radians using our robots takes approximately 0.2 s. Thus, during the portion of the trajectory corresponding to the gap, we specify an acceleration such that $\boldsymbol{b}_3$ is in the same direction of the $\boldsymbol{z}_W$ window axis.

VI. EXPERIMENTAL RESULTS

In this section, we report on the experiments that have been performed at the PERCH lab (Penn Engineering Research Collaboration Hub) at the University of Pennsylvania indoor

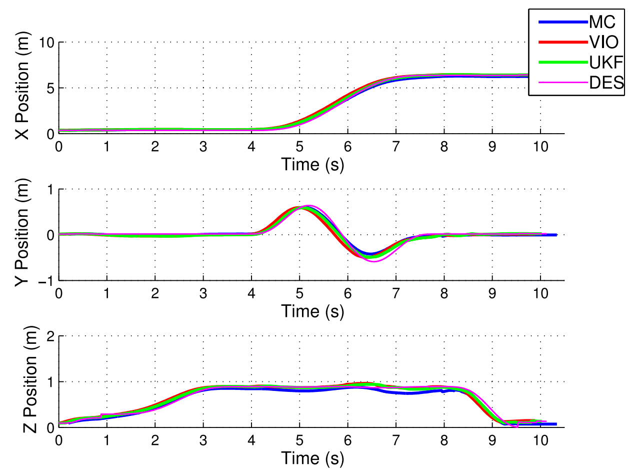

Fig. 8. Ground truth (blue), estimated (green), desired (magenta) Cartesian position for the 6 m slalom trajectory.

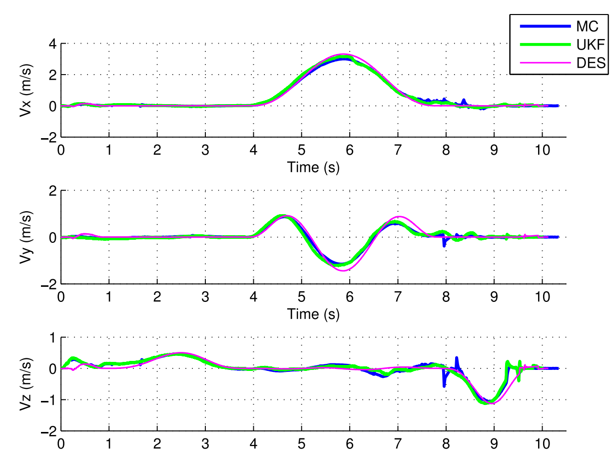

Fig. 9. Ground truth (blue), estimated (green), desired (magenta) Cartesian velocities for the 6 m slalom trajectory.

testbed. The total flying area is a volume of $20 \times 6 \times 4$ m^3. The algorithms for odometry estimation and control are running onboard. A Qualisys[2] motion capture system with 22 Oqus cameras running at 100 Hz is used for ground-truth comparison. We evaluate the two previous aggressive trajectory tasks under different conditions. The VIO module is set to track an average of 20–30 features each frame. A good pose estimation is obtained

[2] http://www.qualisys.com/

TABLE II
ORIENTATION RMSE AND STD IN RADIANS OF THE ESTIMATES COMPARED TO MOTION CAPTURE SYSTEM FOR THE SLALOM CASE

Orientation	Path length (m)	RMSE VIO estimation	RMSE UKF estimation	STD VIO	STD UKF
$\Psi(\boldsymbol{R}, \boldsymbol{R}_d)$	6	6×10^{-2}	3×10^{-2}	5×10^{-2}	3×10^{-2}
	3	1×10^{-3}	1×10^{-3}	9×10^{-3}	1.73×10^{-4}

TABLE III
POSITION AND VELOCITY RMSE AND STD OF THE ESTIMATES COMPARED TO MOTION CAPTURE FOR THE WINDOW CASE

Gap Max. Angle (°)	Max. Acc. (m/s^2)	Max. Ang. Vel. (deg/s)	Cartesian Component	RMSE VIO estimation (m)	RMSE UKF estimation (m)	RMSE estimation (m)	STD position VIO (m)	STD position UKF (m)	STD Velocity (m/s)
0	4	149.2578	x	0.0703	0.0618	0.0805	0.0701	0.0599	0.0795
			y	0.0378	0.0420	0.0507	0.0349	0.0379	0.0496
			z	0.0510	0.0463	0.0516	0.0397	0.0337	0.0514
45	8	257.8085	x	0.0414	0.0480	0.0579	0.0387	0.0446	0.0577
			y	0.0711	0.0754	0.1179	0.0711	0.0753	0.1178
			z	0.0665	0.0764	0.0915	0.0514	0.0633	0.0914
90	15	782.2472	x	0.0527	0.0629	0.0626	0.0390	0.0605	0.0621
			y	0.0415	0.0445	0.1353	0.0387	0.0422	0.1352
			z	0.0469	0.0583	0.1230	0.0364	0.0518	0.1224

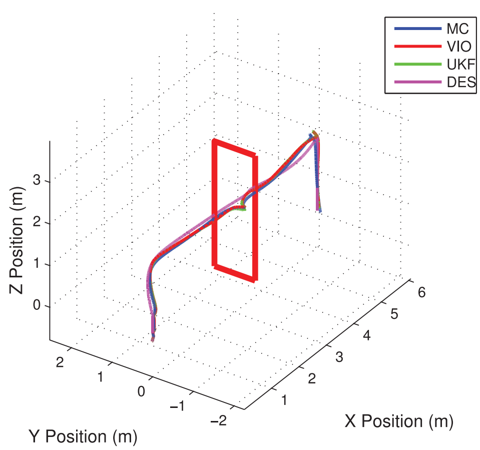

Fig. 10. Ground truth (blue), estimated (green), desired (magenta) trajectories over the window at 90 degrees.

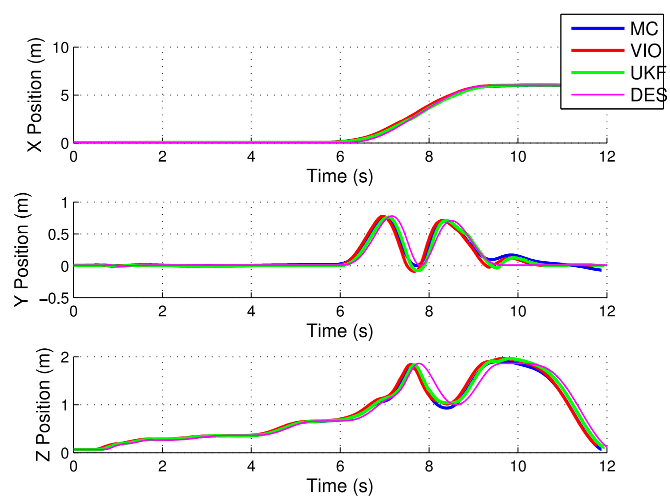

Fig. 11. Ground truth (blue), estimated (green), desired (magenta) Cartesian positions over the window at 90 degrees.

when this number does not drop below 2. The initialization requires around 10 features.

We consider slalom paths around poles that are (a) 3 m and (b) 1.5 m apart requiring the quadrotor to maintain a clearance distance around the poles. The results are reported in Table I. For the first test we reach a maximum speed of 4.5 m/s whereas in the second case 3 m/s with a maximum acceleration of 5 $\mathrm{m/s^2}$ in both cases. Despite only the initial information about obstacles' positions, the vehicle is still able to avoid them. This validates the precision and robustness of our localization strategy. Moreover, it is noticeable that despite the trajectories having different velocities, accelerations, and thus angles, the results are in the same range of values both for Root Mean Square Error (RMSE) and Standard Deviation (STD). For brevity, we do not report on the control error values, which are similar to the estimated ones as visible from Figs. 7, 8, 9. The reader can also notice that the VIO and UKF modules have similar errors with the separation strategy providing the benefits already discussed. The orientation errors in Table II have been computed according to [26]. The attached multimedia material shows that each experiment has been repeated multiple times with different safety distances of up to 0.5 m and with similar performances to the results in Table I.

The results for the narrow gap experiment are reported in Table III. The experiment has been repeated considering different maximum rotation angles on one of the window's axis. The results show that our methodology gives similar results despite different operating conditions and increased level of difficulty induced by a higher values of acceleration, platform angles and larger operating envelope. The traversing gap of the trajectory is encapsulated between the take-off and landing phase that have an average duration of 2.5 s each. The traversing path has a length of 6 m along the x Cartesian axis and a duration of 5 s (see Figs. 10–12, between 5 and 10 s). In all cases, the vehicle reaches an average maximum speed of 4.5 m/s, accelerations of up to 15 m/s^2 and angular rotations of up to 800 deg/s.

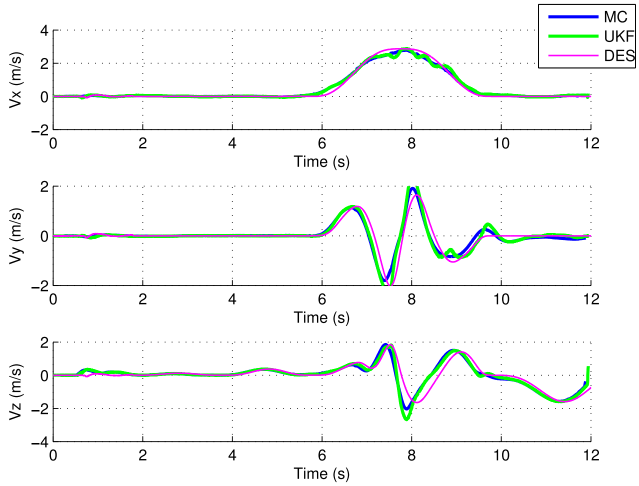

Fig. 12. Ground truth (blue), estimated (green), desired (magenta) Cartesian velocities over the window at 90 degrees.

For brevity, we omit the orientation errors since similar to the slalom task. The reader will notice from the attached multimedia material the successful accomplishment of multiple trials. The acceleration $\boldsymbol{a}_W$ is set to be half of the available thrust.

VII. Conclusion

In this work, we presented the system architecture and methodologies to enable vision-based high speed and aggressive flight with a small scale quadrotor equipped only with a single camera and an IMU as sensors. We demonstrated the algorithm framework for estimation, control and trajectory generation, and experimental performance of our autonomous navigation solution based on camera and IMU with slalom trajectories and flight through narrow windows (gaps) at different angles. We successfully achieve the desired control, state estimation, and trajectory planning through constrained environments, even with limited on-board computational capabilities derived from smartphone grade hardware. Our results reveal the feasibility and correctness of the proposed approach. Specifically we reach speeds of 5 m/s, roll and pitch angles of 90 degrees, accelerations of up to 15 m/s^2 and angular velocities of up to 800 deg/s. Our ongoing work addresses the use of the small baseline, stereo camera for real-time dense mapping to allow obstacle detection and mapping for planning.

Acknowledgment

The authors would like to thank for helpful discussions with Dr. D. Mellinger and his team at Qualcomm Research Philadelphia.

References

[1] T. Tomic *et al.*, "Toward a fully autonomous UAV: Research platform for indoor and outdoor urban search and rescue," *IEEE Robot. Autom. Mag.*, vol. 19, no. 3, pp. 46–56, Sep. 2012.

[2] T. Ozaslan, S. Shen, Y. Mulgaonkar, N. Michael, and V. Kumar, "Inspection of penstocks and featureless tunnel-like environments using micro UAVs," in *Proc. 9th Int. Conf. Field Service Robot.*, Brisbane, Australia, 2013, pp. 123–136.

[3] J. Cacace, A. Finzi, V. Lippiello, G. Loianno, and D. Sanzone, *Aerial Service Vehicles for Industrial Inspection: Task Decomposition and Plan Execution*. Berlin, Germany: Springer-Verlag, 2013, pp. 302–311.

[4] G. Loianno, J. Thomas, and V. Kumar, "Cooperative localization and mapping of MAVs using RGB-D sensors," in *Proc. IEEE Int. Conf. Robot. Autom.*, May 2015, pp. 4021–4028.

[5] F. Forte, R. Naldi, A. Macchelli, and L. Marconi, "Impedance control of an aerial manipulator," in *Proc. Amer. Control Conf.*, Montreal, QC, Canada, Jun. 2012, pp. 3839–3844.

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

[7] N. Michael *et al.*, "Collaborative mapping of an earthquake-damaged building via ground and aerial robots," *J. Field Robot.*, vol. 29, no. 5, pp. 832–841, 2012.

[8] P. Abbeel, "Apprenticeship learning and reinforcement learning with application to robotic control," Ph.D. dissertation, Comput. Sci. Dept., Stanford Univ., Stanford, CA, USA, 2008.

[9] J. H. Gillula, H. Huang, M. P. Vitus, and C. J. Tomlin, "Design of guaranteed safe maneuvers using reachable sets: Autonomous quadrotor aerobatics in theory and practice," in *Proc. IEEE Int. Conf. Robot. Autom.*, May 2010, pp. 1649–1654.

[10] S. Lupashin, A. Schllig, M. Sherback, and R. D'Andrea, "A simple learning strategy for high-speed quadrocopter multi-flips," in *Proc. IEEE Int. Conf. Robot. Autom.*, May 2010, pp. 1642–1648.

[11] D. Mellinger, N. Michael, and V. Kumar, "Trajectory generation and control for precise aggressive maneuvers with quadrotors," *Int. J. Robot. Res.*, vol. 31, no. 5, pp. 664–674, 2012.

[12] E. S. Jones and S. Soatto, "Visual-inertial navigation, mapping and localization: A scalable real-time causal approach," *Int. J. Robot. Res.*, vol. 30, no. 4, pp. 407–430, 2011.

[13] J. Kelly and G. S. Sukhatme, "Visual-inertial sensor fusion: Localization, mapping and sensor-to-sensor self-calibration," *Int. J. Robot. Res.*, vol. 30, no. 1, pp. 56–79, Jan. 2011.

[14] J. A. Hesch, D. G. Kottas, S. L. Bowman, and S. I. Roumeliotis, "Camera-IMU-based localization: Observability analysis and consistency improvement," *Int. J. Robot. Res.*, vol. 33, no. 1, pp. 182–201, 2014.

[15] A. Martinelli, "Visual-inertial structure from motion: Observability and resolvability," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, Nov. 2013, pp. 4235–4242.

[16] S. Weiss, D. Scaramuzza, and R. Siegwart, "Monocular-SLAM-based navigation for autonomous micro helicopters in GPS denied environments," *J. Field Robot.*, vol. 28, no. 6, pp. 854–874, 2011.

[17] S. Shen, N. Michael, and V. Kumar, "Tightly-coupled monocular visual-inertial fusion for autonomous flight of rotorcraft mavs," in *Proc. IEEE Int. Conf. Robot. Autom.*, May 2015, pp. 5303–5310.

[18] S. Shen, Y. Mulgaonkar, N. Michael, and V. Kumar, "Vision-based state estimation and trajectory control towards high-speed flight with a quadrotor," in *Proc. Robot. Sci. Syst.*, Berlin, Germany, 2013.

[19] G. Loianno *et al.*, "Smartphones power flying robots," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, Hamburg, Germany, Sep. 2015, pp. 1256–1263.

[20] E. Mueggler, B. Huber, and D. Scaramuzza, "Event-based, 6-DOF pose tracking for high-speed maneuvers," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, Chicago, IL, USA, Sep. 2014, pp. 2761–2768.

[21] M. Faessler, F. Fontana, C. Forster, and D. Scaramuzza, "Automatic re-initialization and failure recovery for aggressive flight with a monocular vision-based quadrotor," in *Proc. IEEE Int. Conf. Robot. Autom.*, May 2015, pp. 1722–1729.

[22] N. Michael, D. Mellinger, Q. Lindsey, and V. Kumar, "The GRASP multiple micro–UAV test bed," *IEEE Robot. Autom. Mag.*, vol. 17, no. 3, pp. 56–65, Sep. 2010.

[23] T. Lee, M. Leok, and N. H. McClamroch, "Nonlinear robust tracking control of a quadrotor UAV on SE(3)," *Asian J. Control*, vol. 15, no. 2, pp. 391–408, 2013.

[24] D. Mellinger and V. Kumar, "Minimum snap trajectory generation and control for quadrotors," in *Proc. IEEE Int. Conf. Robot. Autom.*, Shangai, China, 2011, pp. 2520–2525.

[25] J. Civera, A. J. Davison, and J. M. M. Montiel, "Inverse depth parametrization for monocular slam," *IEEE Trans. Robot.*, vol. 24, no. 5, pp. 932–945, Oct. 2008.

[26] F. Bullo and A. D. Lewis, *Geometric Control of Mechanical Systems* (Applied Mathematics Series) vol. 49. New York, NY, USA: Springer-Verlag, 2004.