HAL open science

Quadrotor going through a window and landing: An image-based visual servo control approach

Zhiqi Tang, Rita Cunha, David Cabecinhas, Tarek Hamel, Carlos Silvestre

To cite this version:

Zhiqi Tang, Rita Cunha, David Cabecinhas, Tarek Hamel, Carlos Silvestre. Quadrotor going through a window and landing: An image-based visual servo control approach. Control Engineering Practice, 2021, 112, pp.104827. 10.1016/J.CONENGPRAC.2021.104827 . hal-03650973

HAL Id: hal-03650973

https://hal.science/hal-03650973v1

Submitted on 9 May 2023

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Distributed under a Creative Commons Attribution - NonCommercial 4.0 International License

Version of Record: https://www.sciencedirect.com/science/article/pii/S0967066121001040
Manuscript_ce6d70ee8f11f3bf459a03ded538defc

Quadrotor going through a window and landing: An image-based visual servo control approach

Zhiqi Tang, Rita Cunha, David Cabecinhas, Tarek Hamel, Carlos Silvestre

***Abstract*—This paper considers the problem of controlling a quadrotor to go through a window and land on a planar target, the landing pad, using an Image-Based Visual Servo (IBVS) controller that relies on sensing information from two on-board cameras and an IMU. The maneuver is divided into two stages: crossing the window and landing on the pad. For the first stage, a control law is proposed that guarantees that the vehicle will not collide with the wall containing the window and will go through the window with non-zero velocity along the direction orthogonal to the window, keeping at all times a safety distance with respect to the window edges. For the landing stage, the proposed control law ensures that the vehicle achieves a smooth touchdown, keeping at all time a positive height above the plane containing the landing pad. For control purposes, the centroid vectors provided by the combination of the spherical image measurements of a collection of landmarks (corners) for both the window and the landing pad are used as position measurement. The translational optical flow relative to the wall, window edges, and landing plane is used as velocity cue. To achieve the proposed objective, no direct measurements nor explicit estimate of position or velocity are required. Simulation and experimental results are provided to illustrate the performance of the presented controller.**

I. Introduction

Navigation of Unmanned Aerial Vehicles (UAVs) using vision systems has been an important field of research during recent decades. Especially in indoor environments, where GPS is unavailable, a widely adopted alternative sensor suite includes an inertial measurement unit (IMU) and cameras, which are both passive, lightweight, and inexpensive sensors Zingg et al. (2010). Three main solutions have been proposed for navigation using vision in indoor environments: map-based navigation, map-building-based navigation and mapless navigation DeSouza & Kak (2002). The first approach depends on a user-created geometric model or topology map of the environment, e.g. Perspective n Point (PnP), and the second requires the use of sensors to construct their own geometric or topological models, e.g. Simultaneous localization and mapping (SLAM). The mapless visual navigation, in which no global representation of the environment is required and the environment is perceived as the system navigates, can be classified in accordance with the main vision technique or types of clues used during the navigation, which are methods based on optical flow, appearance and feature tracking (DeSouza & Kak (2002)). However, the appearance-based method has the main problems which are to find an appropriate algorithm for the representation of the environment and to define the on-line matching criteria.

Visual servo control is a popular mapless navigation method based on feature tracking, which can be classified in two main categories: Image-based visual servo (IBVS) and Position-based visual servo (PBVS) control. PBVS involves reconstruction of the target pose with respect to the robot thus a 3-D model of the observed object should be known. However in IBVS, the control commands are deduced directly from image features, thus they offer advantages in robustness to camera and target calibration errors, reduced computational complexity, and simple extension to applications involving multiple cameras compared to PBVS methods (Hutchinson et al. (1996)). However, classical IBVS (Chaumette et al. (2016)) suffers from three key problems. First, it is necessary to determine the depth of each visual feature used in the image error criterion independently from the control algorithm. Second, the rigid-body dynamics of the camera ego-motion are highly coupled when expressed as target motion in the image plane. And last, it uses a simple linearized control on the image kinematics that leads to complex non-linear dynamics and is not easily extended to the dynamics.

In order to overcome these problems, a spherical camera geometry can be used, from which the virtual spherical image points can be obtained by transforming the image points on the perspective camera to the view that would be seen by an ideal unified-spherical camera. The passivity-like properties can be recovered for a centroid image feature as long as a spherical camera geometry is used. The novel IBVS algorithms based on spherical image centroids (e.g. applications on hovering an autonomous helicopter Hamel & Mahony (2002), landing a quadrotor on moving platform Hérissé et al. (2012), Serra et al. (2016) and landing a fixed-wing aircraft on the runway Le Bras et al. (2014), Serra et al. (2015), Tang et al. (2018*a*)), do not require accurate depth information for observed image features and overcomes the difficulties associated with the highly coupled dynamics of the camera ego-motion in the

This work was partially supported by the Project MYRG2018-00198-FST of the University of Macau; by the Macao Science and Technology, Development Fund under Grant FDCT/0031/2020/AFJ; by Fundação para a Ciência e a Tecnologia (FCT) through Project UIDB/50009/2020 and PTDC/EEI-AUT/31411/2017. The work of T. Hamel was supported by the DACAR project of the French National Research Agency (ANR). The work of Z. Tang was supported by FCT through Ph.D. Fellowship PD/BD/114431/2016 under the FCT-IST NetSys Doctoral Program.

Z. Tang is with ISR, IST, Universidade de Lisboa, Portugal and I3S-CNRS, Université Côte d'Azur, Nice-Sophia Antipolis, France (zhiqitang@tecnico.ulisboa.pt).

R. Cunha, D. Cabecinhas are with ISR, IST, Universidade de Lisboa, Portugal (rita@isr.tecnico.ulisboa.pt, dcabecinhas@isr.tecnico.ulisboa.pt).

T. Hamel is with I3S-CNRS, Institut Universitaire de France, Nice-Sophia Antipolis, France (thamel@i3s.unice.fr).

C. Silvestre is with Faculty of Science and Technology of the University of Macau, Macao, China and on leave from ISR, IST, Universidade de Lisboa, Portugal (csilvestre@umac.mo).

© 2021 published by Elsevier. This manuscript is made available under the CC BY NC user license
https://creativecommons.org/licenses/by-nc/4.0/

image dynamics.

This paper extends the IBVS control solution based on spherical image centroids to a specific problem of steering a quadrotor to move from one room to a second one by crossing a window and then land on a planar target placed in the second room (see Fig. 1). This application has significant practical interest since many tasks (i.e. search and rescue in an earthquake-damaged building Michael et al. (2012), package delivery using UAVs) require UAVs to land on a final destination or to perform intermediate landings for battery recharge or exchange, or refueling (for larger UAVs) during long missions. The quadrotor is assumed to be equipped with an IMU and two on-board cameras: one forward-looking and another downward-looking. Neither the translational velocity and position of the vehicle nor the location of the target (window and landing pad) are known. In the proposed IBVS control laws, the centroid vectors provided by the combination of the spherical image measurements of a collection of landmarks (corners) from both the window and the landing pad are used as position cue and the translational optical flow relative to the plane containing window and landing pad is used as velocity cue.

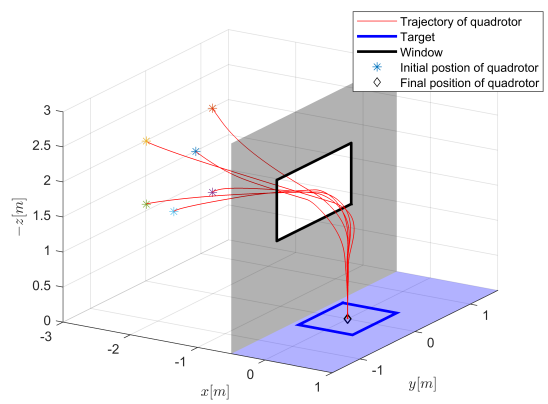

Fig. 1: 3-D plot of the quadrotor trajectories under different initial conditions.

The proposed control strategy draws inspiration from Serra et al. (2016) which combines a centroid-like feature and the translational optical flow to perform exponential landing on a desired spot. This paper considers different control objectives: going through a window and then landing on a desired target. The control law for going through a window ensures that no collision with the wall or windows edges will occur and the vehicle will align with the center line orthogonal to the window, crossing it with non-zero velocity. The control law for the landing is an improvement with respect to the one used in Serra et al. (2016) in which the desired optical flow was chosen constant, leading to a high-gain controller that fails to achieve a perfect landing maneuver. Conversely, the desired optical flow adopted in this work is not constant. It corresponds to the component of the image centroid in the direction orthogonal to the target plane leading to a vanishing desired optical flow when the distance to the target approaches zero and therefore one avoids the high-gain nature of prior work (Serra et al. (2016)).

Following on previous work Tang et al. (2018*b*), which presents the preliminary results with only simulations, this paper presents the following novel contributions: 1) bounded disturbances (e.g. due to wind, and unmodeled dynamics) are included in the dynamics of the system; 2) a complete stability analysis shows that convergence to the desired zero-height equilibrium is guaranteed in all cases and ultimate boundedness of the horizontal position error is guaranteed when landing in the presence of horizontal disturbances; 3) experimental results are provided where the controllers run on an onboard computer together with the image processing for the detection of window and landing pad and for the computation of the translational optical flow.

The body of the paper consists of eight parts. Section II presents the dynamic model, the fundamental equations of motion, and the adopted hierarchical control architecture. Section III introduces the environment and presents the image features that are used in the control laws. Section IV proposes two control laws: one for the landing task in obstacle-free environments and the other for flying through the window. A combination of these two control laws in the practical case is also presented in this section. Section V shows simulation results obtained with the proposed controller. Section VI presents and analyzes the experimental results which validate the proposed controllers. The paper concludes with some final comments in Section VII.

A. Related Work

There are several examples in the literature of recent work dedicated to the problem of flying autonomous vehicles in complex environment using vision systems. In Falanga et al. (2017), Guo & Leang (2020), Loianno et al. (2017), the authors specifically address the problem of going through a window using only a single camera and an IMU. However, estimation of vehicle's position and velocity is required in Falanga et al. (2017), Loianno et al. (2017). Besides, the pose of the window is assumed to be known in Loianno et al. (2017). Although the work in Guo & Leang (2020) directly uses image feature as position cue, estimates of the image depth are still required and the velocity vector is assumed to be known. In general, state estimation adds computational complexity, and the output is often sensitive to image noise and camera calibration errors. The limited work on image-based control approach can be explained by the complexity involved in obtaining sound proofs of convergence and stability.

Landing in complex environments calls for obstacle avoidance capabilities, which are naturally provided by the use of optical flow, a visual feature that draws inspiration from flying insects. Optical flow measures the pattern of apparent motion of objects, surfaces, and edges in a visual scene caused by the relative motion between an observer and a scene (Burton & Radford (1978)). It has been experimentally shown that the neural system of the insects reacts to optic flow patterns to produce a large variety of flight capabilities, such as obstacle avoidance, speed maintenance, odometry estimation,

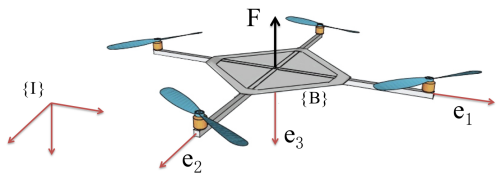

Fig. 2: Reference frames and force for schematic representation of a quadrotor.

wall following and corridor centering, altitude regulation, orientation control and landing (Floreano & Wood (2015), Serres & Ruffier (2017)). Using optical flow as velocity cue and observed feature expressed in terms of an unnormalized spherical centroid, a fully nonlinear adaptive visual servo control design is provided in Mahony et al. (2008). Although estimating the height of the camera above the landing plane was still required, it was the first time that an IBVS control using image measurements for both position and velocity was proposed, going beyond the kinematic model to consider the dynamics. Based on Mahony et al. (2008) and using optical flow, the authors proposed IBVS controllers for landing a quadrotor (Hérissé et al. (2012), Serra et al. (2016)) and landing a fixed-wing aircraft eliminating the need to estimate the height of the vehicle above the ground (Le Bras et al. (2014), Serra et al. (2015), Tang et al. (2018*a*)). Using a distinct paradigm, a novel setup of self-supervised learning based on optical flow was introduced in Ho et al. (2018). Using optical flow, the proposed method learns the visual appearance of obstacles in order to search for a landing spot for micro aerial vehicles.

When compared to related work, this paper proposes simple IBVS controllers applied in sequence to first go through a window and then land on a planar target, using only vision measurements and requiring no estimation of position, velocity, image depth, nor height above the target. The present work also provides rigorous mathematical proofs for stability and robustness in the presence of disturbances, complemented by experimental validation of the proposed controllers.

II. Quadrotor Modeling and Control Architecture

Consider a quadrotor equipped with an IMU and two cameras. To describe the motion of the quadrotor, two reference frames are introduced: an inertial reference frame $\{I\}$ fixed to the earth surface and a body-fixed frame $\{B\}$ attached to the quadrotor's center of mass (see Fig. 2). Let $R = {}^I_B R \in SO(3)$ denote the orientation of the frame $\{B\}$ with respect to $\{I\}$ and let $\xi \in \mathbb{R}^3$ be the position of the origin of the frame $\{B\}$ with respect to $\{I\}$. Let $v \in \mathbb{R}^3$ denote the translational velocity expressed in $\{I\}$ and $\Omega \in \mathbb{R}^3$ the orientation velocity expressed in $\{B\}$. The kinematics and dynamics of the quadrotor vehicle are then described as

$$\begin{cases} \dot{\xi} = v \\ m\dot{v} = -F + mge_3 + \triangle \end{cases} \tag{1}$$

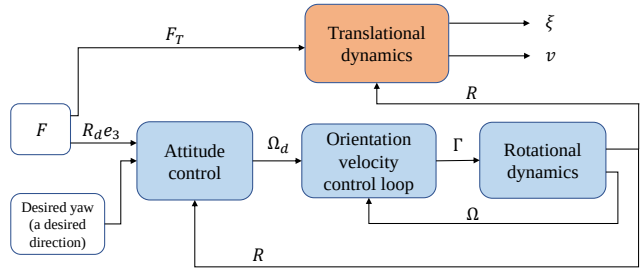

Fig. 3: A hierarchical control design strategy.

$$\begin{cases} \dot{R} = RS(\Omega) \\ \mathbf{I}\dot{\Omega} = -\Omega_\times \mathbf{I}\Omega + \Gamma \end{cases} \tag{2}$$

with g the gravitational acceleration, m the mass of the vehicle and $\mathbf{I}$ its inertia matrix. The matrix $\Omega_\times$ denotes the skew-symmetric matrix matrix associated with the vector product $\Omega_\times x = \Omega \times x$, for any $x \in \mathbb{R}^3$.

The vector $F \in \mathbb{R}^3$ expressed in $\{I\}$ combines the principal non-conservative forces applied to the quadrotor and generated by the four rotors. In quasi-hover conditions one can reasonably assume that this aerodynamic force is always in the direction e_3^b in $\{B\}$, since all the four thrusters are aligned with e_3^b and their contribution predominates over other components. Thus the F in the direction of e_3^b expressed in the inertial frame can be described as follows:

$$F = F_T R e_3 \tag{3}$$

where the scalar F_T represents the total thrust magnitude generated by the four motors. It also represents the unique control input for the translational dynamics.

The term $\triangle$ combines the modeling errors and aerodynamic effects due to the interaction of the rotors wake with the environment causing random wind and dynamic inflow effects (Peters & HaQuang (1988)).

The vector $\Gamma \in \mathbb{R}^3$ expressed in $\{B\}$ is the torque control for the attitude dynamics. It is obtained via the combination of the contributions of four rotors. The invertible linear map between $[F_T \in \mathbb{R}^+, \Gamma \in \mathbb{R}^3]$ and the collection of individual thrusters $[F_{T_1}, F_{T_2}, F_{T_3}, F_{T_4}]$ can be found in Hamel et al. (2002).

A. Control architecture

A hierarchical control design strategy is adopted in this paper (see Fig. 3). This choice is motivated by the natural structure of the system dynamics and its practical implementation (Bertrand et al. (2011), Hérissé et al. (2012)). For the translational dynamics of the quadrotor (Eq. (1)), the force F (Eq. (3)) is used as control input by means of its thrust direction and its magnitude. This constitutes a high-level outer loop for the control design. The thrust F_T is directly the magnitude of the designed force ($F_T = \|F\|$) and the desired attitude R_d (partly obtained by the desired direction $R_d e_3 = \frac{F}{\|F\|}$ complemented by a desired yaw) can then be reached by considering the body's angular velocity Ω as an intermediary control input, which constitutes again a desired

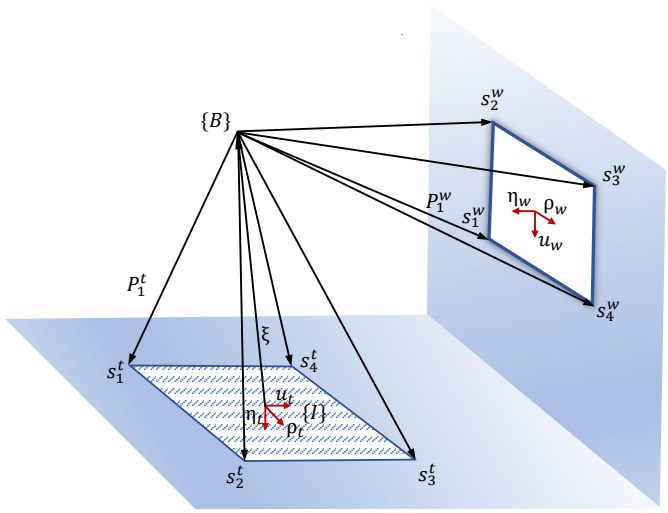

Fig. 4: Landing plane and window plane.

angular velocity for the fully actuated orientation dynamics (Eq. (2)) via the high gain control torque Γ. The stabilisation of the orientation dynamics is not the subject of this paper and it is assumed that a suitable low level robust stabilising control is implemented, that satisfactorily regulates the attitude error with a fast dynamics.

III. Environment and Image Features

In this section adequate image features in relation to the considered tasks are derived and all required assumptions regarding the environment and the setup are established.

Assumption 1. *A downward-looking camera and a forward-looking camera are attached to the center of mass of the vehicle. The downward-looking camera reference frame coincides with the body-fixed frame {B}. The rotation matrix from the forward-looking camera reference frame to the body frame* ${}^{B}_{C}R \in SO(3)$ *is known.*

Assumption 2. *The angular velocity* Ω *is measured and the orientation matrix* R *of {B} with respect to {I} is obtained by external observer-based IMU measurements. This allows to represent all image information and the system dynamics in the inertial frame.*

Assumption 3. *The landing target lies on a textured plane which is called target plane. Its normal direction* $\eta_t \in \mathbb{S}^2$ *in the inertial frame is known (typically* $\eta_t \approx e_3$*).*

Assumption 4. *The target window has a rectangle shape and lies on a textured wall which is called window plane. Its width* r_w *is known but its normal direction* $\eta_w \in \mathbb{S}^2$ *is unknown.*

Both landing plane and window plane are placed in the environment, as shown in Figure 4. It assumed that the vehicle is able to recognize the landing pad and the window from landmarks on the pad and from corners and edges of the window respectively. The background texture on both landing plane and window plane are also exploited to obtain information about the vehicle's velocity with respect to the planes and also to avoid collisions with the wall and the window's edges.

For any initial position (along with any initial velocity) outside the room containing the landing pad, the main objective is to design a feedback controller resorting only to image features that can ensure automatic landing of the vehicle without any collision.

A. Image features on the landing plane

The target on the landing plane is depicted in Figure 4. The axes of $\{I\}$ are given by (u_t, ρ_t, η_t), where $\rho_t = \eta_t \times u_t$, and the origin of $\{I\}$ is placed at the center of the landing pad. As shown in Figure 4, $s_i^t \in \mathbb{R}^3$ denotes the position of ith marker (or a corner) of the landing pad relative to the inertial frame expressed in $\{I\}$. Note that $\eta_t^\top s_i^t = 0$. Define the position vector of ith marker of the target relative to $\{B\}$ as

$$P_i^t = s_i^t - \xi. \tag{4}$$

The position of the vehicle relative to the center of the landing pad is defined as

$$\xi_t = -\frac{1}{n_t}\sum_{i=1}^{n_t} P_i^t = \xi - \frac{1}{n_t}\sum_{i=1}^{n_t} s_i^t \tag{5}$$

where n_t is the number of observed markers on the landing pad and $\frac{1}{n_t}\sum_{i=1}^{n_t} s_i^t$ is a constant vector. This sum is zero when all markers are in the camera field of view.

Using the spherical projection model for a calibrated camera, the spherical image points of landing pad's markers can be expressed as

$$p_i^t = \frac{P_i^t}{\|P_i^t\|} = \frac{s_i^t - \xi}{\|s_i^t - \xi\|} \tag{6}$$

It can be obtained from the 2D pixel locations (X_i^t, Y_i^t) of the camera image, such that

$$p_i^t = R\frac{\bar{p}_i^t}{||\bar{p}_i^t||}, \quad \text{with} \quad \bar{p}_i^t = A^{-1}\begin{bmatrix} X_i^t \\ Y_i^t \\ 1 \end{bmatrix}. \tag{7}$$

The matrix A^{-1} in the above equation is the camera's intrinsic parameters that transforms image pixel to perspective coordinates $\bar{p}_i^t$. Note that expressions (6) and (7) are the same and hence p_i^t does not depend on the orientation.

The visual feature used for the landing task is the the centroid of the observed visual feature.

$$q_t := -\frac{1}{n_t}\sum_{i=1}^{n_t} p_i^t = -R\left(\frac{1}{n_t}\sum_{i=1}^{n_t}\frac{\bar{p}_i^t}{||\bar{p}_i^t||}\right) \tag{8}$$

which is the simplest image feature that encodes all information about the position of the vehicle with respect to the landing plane. It is not necessary to match observed image points with desired features as required in classical image based visual servo control. Besides, the calculation of the image centroid is highly robust to pixel noise, and easily computed in real-time in the camera frame and then derotated. This ensures that q_t is invariant to any orientation motion (Hamel & Mahony (2002)).

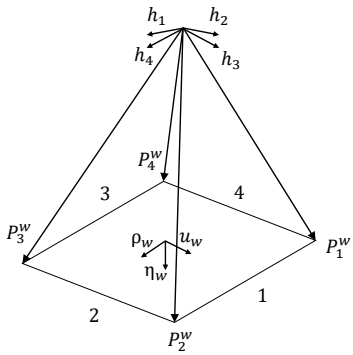

Fig. 5: Window plane and unit directions h_i normal to the planes defined by the origin of camera frame and the ith window edge.

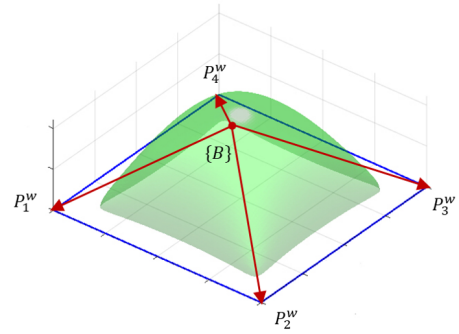

Fig. 6: The green volume represents the region $\mathcal{W}$ defined by inequality (17) which excludes the window edges.

B. Image features on the window plane

As shown in the Figure 4, a rectangular window is placed on a textured wall. Its corners and edges are assumed to be recognized in camera images. Both information are combined together to extract the normal direction η_w and provide the feedback information used in the controller.

Consider first the the windows corners and let $s_i^w \in \mathbb{R}^3$ denote the position of ith corner of the window expressed in $\{I\}$. Define the position vector of ith corner of the window relative to $\{B\}$ as

$$P_i^w = s_i^w - \xi. \tag{9}$$

From there, one can deduce the position of the vehicle with respect to the window's center:

$$\xi_w = -\frac{1}{n_w}\sum_{i=1}^{n_w} P_i^w = \xi - \frac{1}{n_w}\sum_{i=1}^{n_w} s_i^w, \tag{10}$$

with n_w (typically $n_w = 4$) number of the window's corners and $\frac{1}{n_w}\sum_{i=1}^{n_w} s_i^w$ constant vector.

Similarly to Section III-A and recalling that the forward-looking camera is used to detect the window, the spherical image points of the corners of the window are exploited:

$$p_i^w = \frac{P_i^w}{\|P_i^w\|} = R\,{}^B_C R\,\frac{\bar{p}_i^w}{||\bar{p}_i^w||}, \tag{11}$$

with $\bar{p}_i^w$ the perspective coordinates of the ith window's corner, leading the following centroid:

$$q_w(t) := -\frac{1}{n_w}\sum_{i=1}^{n_w} p_i^w(t) = -R\,{}^B_C R\left(\frac{1}{n_w}\sum_{i=1}^{n_w}\frac{\bar{p}_i^w}{||\bar{p}_i^w||}\right), \tag{12}$$

where ${}^B_C R$ is the rotation matrix from the forward-looking camera reference frame to the body frame.

Now, to extract the normal direction η_w, recall that the axes representing the window are given by (η_w, ρ_w, u_w), with $\rho_w = u_w \times \eta_w$ (see Figure 5). Using the image of ith line and exploiting the fact that the window has a rectangular shape, it is straightforward to get the directions u_w and ρ_w and consequently η_w. As described in Mahony & Hamel (2005), in the binormalized Euclidean Plucker coordinates, the ith line can be represented by its unit direction u_w (resp. ρ_w) and the unit direction h_i, which is normal to the plane defined by the origin of the camera/body-fixed frame and the ith line. The unit vector h_i can be obtained directly from the images of lines which can be identified using a convenient line detection technique, such as the Hough transform. Using the fact that lines 1 and 3 (resp. lines 2 and 4) are parallel in the inertial frame, one deduces the measure of the direction u_w (resp. ρ_w) from the following relationships:

$$\rho_w = \pm\frac{h_1 \times h_3}{\|h_1 \times h_3\|} \tag{13}$$

$$u_w = \pm\frac{h_2 \times h_4}{\|h_2 \times h_4\|}. \tag{14}$$

Then the normal vector to the window plane is directly obtained by

$$\eta_w = \pm\frac{u_w \times \rho_w}{\|u_w \times \rho_w\|} \tag{15}$$

and the sign of equation (15) is chosen such that the condition $\eta_w^\top q_w(0) < 0$, with $q_w(t)$ the image centroid of window's corners in equation (12).

To exploit the image of window's edges, defining the vector from the vehicle to the closest point on window's edges as $L^e \in \mathbb{R}^3$, its direction $l^e = \frac{L^e}{\|L^e\|}$ can be obtained from the camera

$$l^e = \{l_i^e : \max\{|\eta_w^\top l_i^e|\}, i = \{1,2,3,4\}\} \tag{16}$$

where

$$l_i^e = \pm(h_i \times \rho_w), i = \{1,3\}, \quad l_i^e = \pm(h_i \times u_w), i = \{2,4\}$$

are the directions from the vehicle to the nearest point on each edge i.

Form now on, it is able to derive the required information achieving the double goal of going through the window in the meanwhile avoiding collision with the window edges and wall. We first define the safety region $\mathcal{W}$ such that

$$\mathcal{W} := \{\xi_w : \|q_w(\xi_w)\| \leq \epsilon\}, \tag{17}$$

where $\epsilon > 0$ is chosen such that $\forall \xi_w \in \mathcal{W}$, the condition $\|\xi_w\| < \frac{r_w}{2} - \epsilon$ also holds, implying that the region $\mathcal{W}$ does not contain the window edges (see Fig. 6).

From there, the chosen visual feature that encodes all required information about the position of the vehicle with respect to the window is:

$$\bar{q}_w := -\frac{1}{n_w}\sum_{i=1}^{n_w} p_i^w \left[\alpha_w(t)\frac{1}{\eta_w^\top p_i^w} + (1-\alpha_w(t))\frac{\eta_w^\top l_e}{\eta_w^\top p_i^w}\right], \quad (18)$$

where $\alpha_w(\|q_w(\xi_w)\|)$ is a weight function ensuring the continuity of $\bar{q}_w$. It is defined as follows:

$$\alpha_w(t) = \begin{cases} 0 & \text{, if } \|q_w\| \leq \epsilon \ (\xi_w \in \mathcal{W}) \\ \frac{1}{\delta}(\|q_w\| - \epsilon), & \text{, if } \epsilon < \|q_w\| < \epsilon + \delta \\ 1 & \text{, if } \|q_w\| \geq \epsilon + \delta, \end{cases} \quad (19)$$

with δ an arbitrary small positive constant. Since $\eta_w^\top l^e = \frac{\eta_w^\top L^e}{\|L^e\|} = \frac{\eta_w^\top P_i^w}{\|L^e\|}$, $\bar{q}_w$ can be expressed in terms of the unknown distance d_o and d_e:

$$\bar{q}_w(t) = \alpha_w(t)\frac{\xi_w(t)}{d_o(t)} + (1-\alpha_w(t))\frac{\xi_w(t)}{d_e(t)} \quad (20)$$

where $d_o := \eta_w^\top P_i^w = -\eta_w^\top \xi_w$ is the distance from the camera to the wall and $d_e := \|L^e\| = \sqrt{d_o^2 + \|\pi_{\eta_w} L^e\|^2}$ represents the distance from the camera to the closest window's edge.

C. Image Kinematics and Translational Optical Flow

The kinematics of any observed points on the landing plane (including markers of the the landing pad) can be written as:

$$\dot{P}^t = -\dot{\xi} = -v \quad (21)$$

where P^t expressed in $\{I\}$ denotes any point on the textured ground of the landing plane. So the kinematics of the corresponding image point $p^t = \frac{P^t}{\|P^t\|}$ can be expressed as

$$\dot{p}^t = -\pi_{p^t}\frac{v}{\|P^t\|}. \quad (22)$$

with

$$\pi_y := I_3 - yy^\top \geq 0,$$

the orthogonal projection operator in $\mathbb{R}^3$ onto the 2-dimensional vector subspace orthogonal to any $y \in \mathbb{S}^2$. Let d_t be the height of the vehicle above the landing plane:

$$d_t := \eta_t^\top P^t = \eta_t^\top P_i^t = -\eta_t^\top \xi_t, \quad (23)$$

then equation (22) can be rewritten as

$$\dot{p}^t = -\cos\theta^t \pi_{p^t}\phi_t(t) \quad (24)$$

where $\cos\theta^t = \frac{d_t}{\|P^t\|} = \eta_t^\top p^t$ and ϕ_t is the translational optical flow:

$$\phi_t(t) = \frac{v(t)}{d_t(t)} \quad (25)$$

which is the ideal image velocity cue that can be complemented with the centroid information for designing a pure IBVS controller to perform the landing task.

The translational optical flow ϕ_t can be obtained by integrating $\dot{p}^t$ (24) over a solid angle S^2 of the sphere around the normal direction η_t to the landing plane. It can be shown that the average of the optical flow is calculated as in Hérissé et al. (2012):

$$\phi_t(t) = -(R_t\Lambda^{-1}R_t^\top)\int\int_{S^2}\dot{p}^t dp^t, \quad (26)$$

where matrix Λ is a constant diagonal matrix depending on parameters of the solid angle S^2, and R_t represents the orientation matrix of the landing plane with respect to the inertial frame. Since $\{I\}$ is chosen coincident with the target frame one has $R_t = I_3$.

In practice, the optical flow is first measured in the camera frame from the 2-D optical flow $\dot{\bar{p}}_t$ obtained from a sequence of images using the Lucas-Kanade algorithm and then derotated (see Hérissé et al. (2012) for more detail). Note however that computing the optical flow from (26) or directly from $\dot{\bar{p}}_t$ in the camera frame and then derotating it, the result is theoretically the same and does not depend on the measured Ω nor on the estimated R.

Similarly, the kinematics of any observed points on the window plane can be written in the inertial frame as

$$\dot{P}^w = -v \quad (27)$$

where P^w expressed in $\{I\}$ denotes the position of a point on the textured wall of the window plane with respect to $\{B\}$ expressed in $\{I\}$, not to be confused with P_i^w in Eq. (9), which is the position of the ith corner of the window with respect to $\{B\}$ and also expressed in $\{I\}$. So the kinematics of the corresponding image point $p^w = \frac{P^w}{\|P^w\|}$ can be written as

$$\dot{p}^w = -\cos\theta^w \pi_{p^w}\frac{v}{d_o} \quad (28)$$

with $\cos\theta^w = \frac{d_o}{\|P^w\|} = \eta_w^\top p^w$. Analogously to the previous case, the translational optical flow with respect to the textured wall $\frac{v}{d_o}$ can be obtained from the integral of $\dot{p}^w$ along the direction η_w over a solid angle.

Now, to achieve the goal that the vehicle is going through the window smoothly, the translational optical flow with respect to the closest window's edge is also used. The kinematics of any observed points on the closest window's edge is

$$\dot{P}^e = -v \quad (29)$$

where P^e denotes the position of a point on the the closest edge from the window. The kinematics of the corresponding image point $p^e = \frac{P^e}{\|P^e\|}$ can be written as

$$\dot{p}^e = -\cos\theta^e \pi_{l^e}\frac{v}{d_e} \quad (30)$$

with $\cos\theta^e = \frac{d_e}{\|P^e\|} = l^{e\top}p^e$. The translational optical flow with respect to the closest window edge, $\frac{v}{d_e}$, can be obtained from the integral of $\dot{p}^e$ along the direction l^e over a solid angle.

Analogously to (18), the translational optical flow used for going through the window is the convex combination of the translational optical flow with respect to the textured wall and to the closest window edge, respectively:

$$\phi_w = \alpha_w(t)\frac{v(t)}{d_o(t)} + (1-\alpha_w(t))\frac{v(t)}{d_e(t)} \quad (31)$$

with $\alpha_w(t)$ defined already by (19).

IV. Controller design

A. Landing in obstacle free environment

Theorem 1. *Consider the system (1) in the nominal case ($\triangle \equiv 0$) subjected to the following feedback control:*

$$F_t = K_p^t q_t + K_d^t \phi_t + mge_3. \tag{32}$$

with $K_p^t = k_{p_{1,2}}^t \pi_{\eta_t} + k_{p_3}^t \eta_t \eta_t^\top$ and $K_d^t = k_{d_{1,2}}^t \pi_{\eta_t} + k_{d_3}^t \eta_t \eta_t^\top$ two constant positive definite matrices. If for any initial condition such that $d_t(0) = -\eta_t^\top \xi_t(0) \in \mathbb{R}^+$, then the following assertions hold $\forall t \geq 0$:

1) *the height $d_t(t) = -\eta_t^\top \xi_t(t) \in \mathbb{R}^+$ and its derivative $\dot{d}_t(t) \in \mathbb{R}$ are well defined and uniformly bounded $\forall t$ and converge to zero asymptotically,*
2) *the acceleration $\dot{v}(t)$ and the states $(\xi_t(t), v(t))$ are bounded and converge asymptotically to zero.*

Proof. See Appendix A. □

Proposition 1. *Consider the system (1) in which $\triangle$ and $\dot{\triangle}$ are bounded.*

1) *If the perturbation $\triangle$ is such that:*

$$\triangle = \pi_{\eta_t} \triangle, \text{ or equivalently } \eta_t^\top \triangle(t) = 0, \ \forall t \geq 0,$$

then, for any initial condition such that $d_t(0) = -\eta_t^\top \xi_t(0) \in \mathbb{R}^+$, direct application of the feedback control (32) ensures that: i) Item 1 of Theorem 1 holds, ii) $\dot{v}(t)$ and $v(t)$ are bounded and converging asymptotically to zero, and finally iii) $\|\pi_{\eta_t} \xi_t\|$ is ultimately bounded by Δ_ξ, solution of $\|\pi_{\eta_t} q_t\| = \frac{\|\triangle\|_{\max}}{k_{d_{1,2}}^t}$.

2) *If $\eta_t^\top \triangle(t) \neq 0$, then, for any initial condition such that $d_t(0) = -\eta_t^\top \xi_t(0) \in \mathbb{R}^+$, the following slightly modified feedback control:*

$$F_t = K_p^t q_t + K_d^t (\phi_t - \eta_t \phi_t^*) + mge_3 \tag{33}$$

with $\phi_t^* \geq \frac{1}{k_{d_3}^t} |\eta_t^\top \triangle(t)|_{\max}$, ensures that the above i) and ii) assertions hold and guarantees that ξ_t is bounded.

Proof. See Appendix B. □

Remark 1. *The focus of the above proposition is on robustness and adaptation of the controller with respect to the bounded perturbation $\triangle$. It is introduced particularly to show robustness of the proposed control law with respect to bounded perturbations in the plane orthogonal to η_t and, in the interest of a less complicated presentation, a slightly modified version of the control law (32) is introduced in (33) to be able to analyse the robustness of the closed loop system with respect to any bounded disturbance.*

B. Going through the center of the window

To accomplish the goal of going through the window, while avoiding the wall and window edges, the following control law is proposed

$$F_w = \sigma(q_w)(k_p^w \pi_{\eta_w} \bar{q}_w + k_d^w \pi_{\eta_w} \phi_w + k_\phi^w \eta_w ({\eta_w}^\top \phi_w - \phi_w^*) + mge_3), \tag{34}$$

with k_p^w, k_d^w and k_ϕ^w positive gains, $\phi_w^* > 0$ and

$$\sigma(q_w) = \begin{cases} 0 & \text{, if } \eta_w^\top q_w \geq 0 \\ 1 & \text{, if } \eta_w^\top q_w < 0, \end{cases} \tag{35}$$

which indicates that when the vehicle already crossed the window ($d_o \leq 0$), $F_w = 0$. Note that when $\eta_w^\top q_w < 0$, the resulting closed-loop system can be written as

$$\begin{cases} \dot{\xi}_w = v \\ \dot{v} = -k_p^w \pi_{\eta_w} \dfrac{\xi_w}{d_w} - k_d^w \pi_{\eta_w} \dfrac{v}{d_w} - k_\phi^w \eta_w ({\eta_w}^\top \dfrac{v}{d_w} - \phi_w^*) + \triangle, \end{cases} \tag{36}$$

The unknown term d_w is a convex combination of the unknown distances d_o and d_e:

$$\frac{1}{d_w} = (\alpha_w \frac{1}{d_o} + (1 - \alpha_w) \frac{1}{d_e}) \tag{37}$$

which is deduced from (20) and (31) according to the definition of α_w (19):

$$d_w = \begin{cases} d_e, & \text{if } \|q_w\| \leq \epsilon \ (\xi_w \in \mathcal{W}) \\ \frac{d_o d_e}{\alpha_w d_e + (1-\alpha_w) d_o}, & \text{if } \epsilon < \|q_w\| < \epsilon + \delta \\ d_o. & \text{if } \|q_w\| \geq \epsilon + \delta \end{cases} \tag{38}$$

Remark 2. *Note that the unknown time varying distance d_w involved in the closed-loop system is due to the use of feedback information $\bar{q}_w = \frac{\xi_w}{d_w}$ and $\phi_w = \frac{v}{d_w}$ in the control law. It is the key feature to achieve the double goal of avoiding collision with the wall and window edges as well, while ensuring the main task of going through the center of the window. When the vehicle approaches the wall or window edges outside the region $\mathcal{W}$, $d_w = d_o$. If d_o is decreasing then the motion in the orthogonal direction to the wall is highly damped while the region $\mathcal{W}$ is highly attractive. In practice, this leads to a bounded high gain in the feedback control that prevents collision. When the vehicle is inside the region $\mathcal{W}$, $d_w = d_e$. This later is lower bounded by a positive constant so that the vehicle is able to go through the center of the window with a non-zero velocity. More details of analysis will be shown below.*

Proposition 2. *Consider the system (1) with the control input given by (34). If the positive gains k_p^w, k_d^w and k_ϕ^w are such that $\frac{{k_d^w}^2}{k_p^w} > \frac{r_w}{2}$ and for any arbitrary small $\epsilon > 0$, the chosen ϕ_w^* satisfies:*

$$\phi_w^* > \frac{|\eta_w^\top \triangle(t)|_{\max}}{k_\phi^w} + \epsilon, \ \forall t \geq 0, \tag{39}$$

then for any initial condition satisfying $d_w(0) \in \mathbb{R}^+$ and as long as $\sigma(q_w(t)) = 1$, the following assertions hold $\forall t \geq 0$:

1) *there exists a finite time $t_w \geq 0$ at which the vehicle enters the region $\mathcal{W}$ ($\|q_w(t_w)\| \leq \epsilon$) and remains there, while $d_w(t) \geq d_o(t) \in \mathbb{R}^+, \ \forall t < t_w$,*
2) *there exists a finite time $t_{\lim} > t_w$ at which the vehicle crosses the window $d_o(t_{\lim}) = 0$, with strictly negative velocity $\dot{d}_o(t_{\lim})$ such that the vehicle is inside the region $\mathcal{W}$ ($\|q_w(t)\| \leq \epsilon$) for all $t \in [t_w, t_{\lim})$.*

Proof. See Appendix C. □

C. Application Scenario

The double goal of crossing the window and landing on the landing pad can be achieved by simply applying the control laws F_w and F_t in sequence, with an adequate trigger to switch from F_w to F_t. Taking the limitation of the cameras' field of view into the consideration, there will be four different modes during the full process of going through a window and landing on the pad. When $t \in [T_1, T_2)$, $mode = 1$ and F_w (34) is active. When the vehicle approaches to the center of the window, the on-board camera loses the full image of the window and $mode$ changes to 2. When $t \in [T_2, T_3)$, $mode$ 2 is active and the open-loop control $\eta_w |\eta_w^\top F_w(T_2^-)|$ is applied, where T_2^- is the last time instance before the camera loses the image of the window. At the time instance $t = T_3$, when the downward-looking camera detects the landing pad, the $mode$ changes to 3 and the control law F_t (32) is applied when $t \in [T_3, T_4)$. At time instance $t = T_4$, the vehicle is already close to the center of the landing target and it is safe to slowly shutdown the quadrotor motors. In order to avoid inadequate behaviors, the switch from $mode$ 2 to 3 is only triggered once. Moreover, in practice, due to the limitation of camera's field of view, the initial errors should not be large and should converge to zero fast enough, thereby allowing the vehicle to almost align with the center of the window before switching to $mode$ 2. Additionally, the position of the landing target should be close enough to the window so that the quadrotor is able to timely detect the landing target after it goes through the window. The switching between different modes is based on the combination of selected frames from both the downward-looking and forward-looking on-board cameras obtained in the experiments. The detail on the adopted procedure is described in Section VI.

V. Simulation Results

In this section, simulation results are presented to illustrate the behavior of the closed-loop system using the proposed controller. A high-gain inner-loop controller is used to control the attitude dynamics (Tang et al. (2015)). It generates the torque inputs in order to stabilize the orientation of the vehicle to a desired one defined by the desired thrust direction $R_d e_3$, which is provided by the outer-loop image-based controller, and the desired yaw chosen to align the forward-looking camera with direction orthogonal to the wall.The control algorithm is tested with different initial conditions, always starting from a position outside the room containing the target (see Fig. 1). The initial velocity of the quadrotor is $v(0) = [0\ 0\ 0]^\top$, and the gains are chosen as $K_p^t = \text{diag}[4\ 4\ 1.75]$, $K_d^t = 4I_3$, $k_d^w = 0.8$, $k_p^w = 1$, $k_\phi^w = 1$ and $\phi_w^* = 0.3$

As shown in Fig. 1, with different initial positions the quadrotor successfully avoids the wall and window, goes through the center of the window, and then lands on the center of the landing target. Figures 7-15 show in detail the time evolution of quadrotor's state variables, virtual input, and image features for the initial position $\xi(0) = [-2\ 0.1\ -1.82]^\top$. The time evolution of the active mode is also specified. In mode 1, the quadrotor is approaching the window; in mode 2, it is crossing the window with no image cues; in mode 3, it starts detecting the landing pad and transitions to the landing maneuver; and finally in mode 4, the motors are shutdown.

Figure 7 shows the time evolution of the vehicle's position and the dashed lines are the coordinates of window's center. From Figure 7, one can see the quadrotor first converges to the center line of the window and then converges to the center point of the target. Fig. 8 shows the time evolution of the vehicle's velocity. The virtual control input F is shown in Fig. 9. The angular velocity of the quadrotor is depicted in Fig. 10 and Fig. 11 depicts the time evolution of Euler angles, which indicates a good compromise in terms of time-scale separation between the outer-loop and inner-loop controller. Figures 12 and 13 show the translational optical flow used for going through the window in mode 1 and for landing in mode 3, respectively. The evolution of image features of $\bar{q}_w$ and q_t are depicted in Figures 14 and 15, respectively. We can see that the image features $\bar{q}_w$ and q_t approach to the desired values $[-1\ 0\ 0]^\top$ and $[0\ 0\ 0]^\top$, respectively, before the on-board cameras lose the image information.

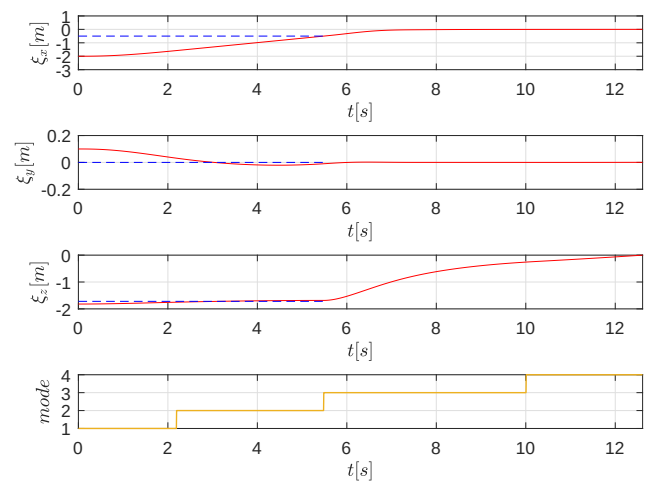

Fig. 7: Evolutions of the quadrotor's position ξ and the mode

VI. Experiments

A. Experimental setup

In order to set up the experiment, a movable wall was used to divide the testing space into two smaller compartments and a landing pad was placed on the ground of the second one. The partition wall contains a rectangular window and is textured as a brick wall to provide the background optical flow, as shown in Fig. 18. The vehicle used for the experiments is an Asctec Pelican quadrotor (Fig. 16) with weight 1676g and the arm length from the center of mass to each motor is 20cm. The available commands are thrust force and attitude which are derived from the force F provided by the outer-loop controller (32) (respectively (34)) and the desired yaw angle. The quadrotor is equipped with two wide-angle cameras, one pointing towards the ground and another is facing the forward direction, pointing at the wall. Recalling Assumption 1, the downward-looking camera reference frame coincides with the

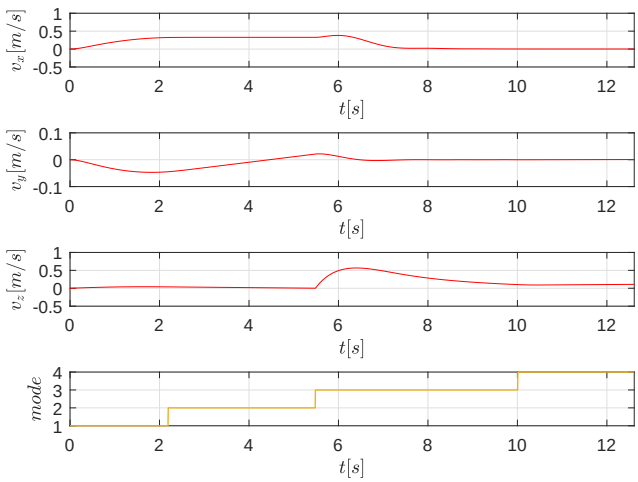

Fig. 8: Evolutions of the quadrotor's velocity v

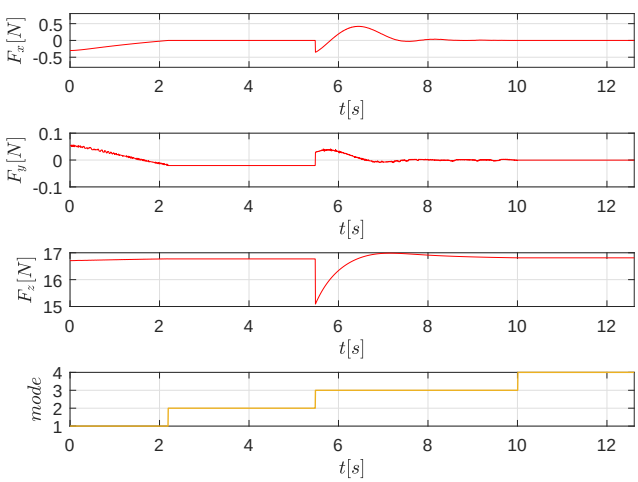

Fig. 9: Evolutions of the virtual control input F

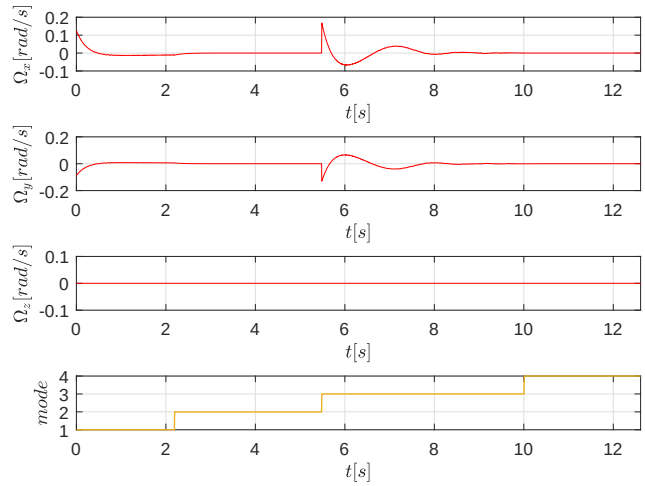

Fig. 10: Evolution of angular velocity Ω

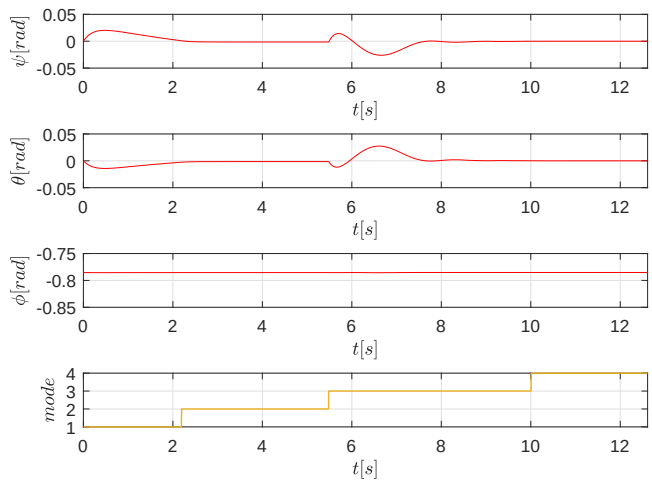

Fig. 11: Evolution of Euler angles

vehicle's body fixed frame and the rotation matrix from the forward-looking camera reference frame to the body frame is ${}^{B}_{C}R = R_Z(-\frac{\pi}{4})R_X(\frac{\pi}{2})$. These two cameras are uEye UI-122ILE models featuring a 1/2-in sensor with global shutter which operate at a resolution of 752×480 pixel at 50 frames per second and are provisioned with 2.2-mm lenses.

In the experiments, a rapid prototyping and testing architecture are used in which a MATLAB/Simulink environment integrates the sensors and the cameras, the control algorithm and the communication with the vehicle. The controller is developed and tuned on a MATLAB/Simulink environment and C code is generated and compiled to run onboard the vehicle as a final step. The onboard computer (a 4-core Intel i7-3612QE at 2.1GHz, named AscTec Mastermind) is responsible for running in Linux three major software components that provide:

1) interface with the camera hardware, image acquisition, feature detection and optical flow computation;
2) computation of the vehicle force references from the image features, translational optical flow, and angular velocity and rotation matrix estimates provided by the IMU;
3) interface with microprocessor, receiving IMU data and sending force references to the inner-loop controller.

A Python program running on the onboard computer performs detection of the window, detection of landing target, and optical flow computation using the OpenCV library. ARUCO markers, for which built-in detection functions exist in the OpenCV library, are used to define the landmarks on the landing pad. In order to fit the camera's field of view during the full process of landing, the landmarks are composed by 4 groups of ARUCO markers and in each group there are 4 ARUCO markers with same border size but different identifier (id) as shown in Fig. 17. When the camera is far

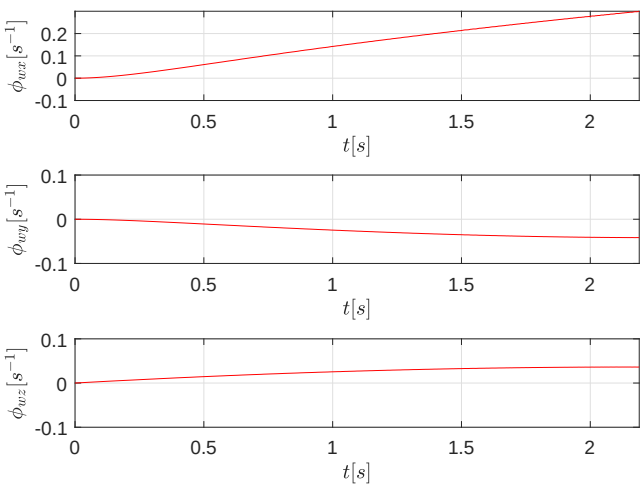

Fig. 12: Translational optical flow using for going through the window during mode 1.

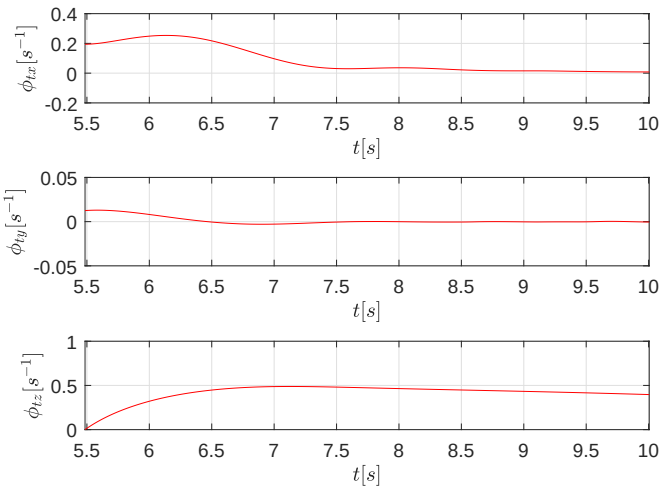

Fig. 13: Translational optical flow using for landing during mode 3.

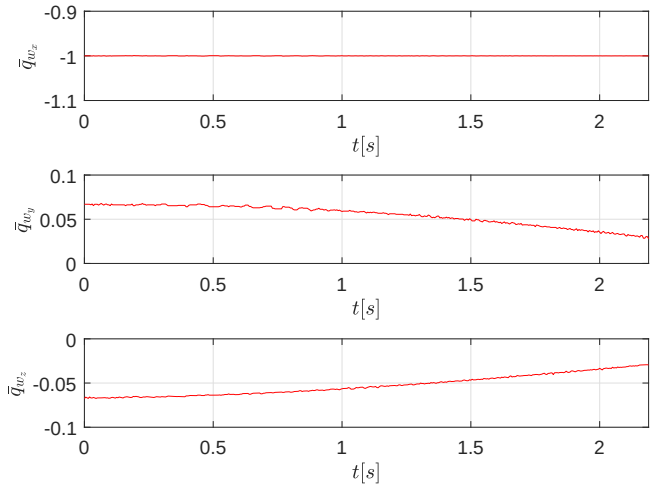

Fig. 14: Image feature $\bar{q}_w$ during mode 1.

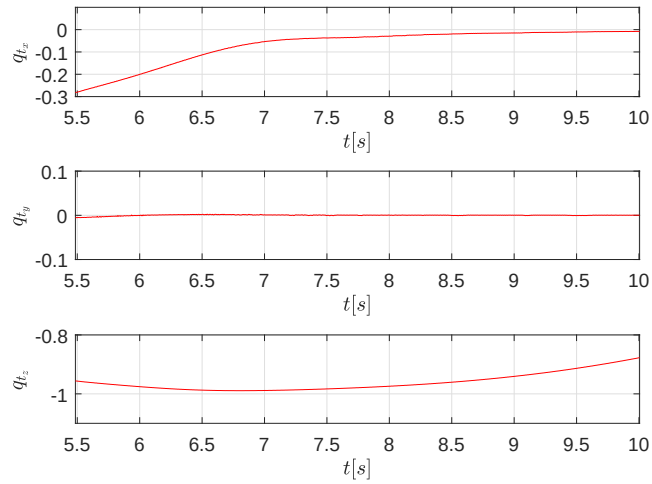

Fig. 15: Image feature q_t during mode 3.

Fig. 16: Asctec Pelican quadrotor.

away from the markers, the group of larger markers can be seen and when the camera is near the ground, only the smaller group of the landmarks will be shown in the field of view. The rectangular window shape is detected using the library code originally developed for ARUCO marker's border detection. The detected window frame (in green) and the window's coordinate system overlayed on the image are show in Fig. 18 (1), (2), and (3). The translational optical flow is also computed onboard. The computation is based on the conventional image plane optical flow field provided by a pyramidal implementation of the Lucas-Kanade algorithm. The detailed description of the computation can be found in Serra et al. (2016). The small vectors represented in Fig. 19 represent the translational optical flow of the image pixels.

In order to provide ground truth measurements and evaluate the performance of the proposed controller, a VICON motion capture system (VICON (2014)) which comprises 12 cameras is used together with markers attached to the quadrotor,

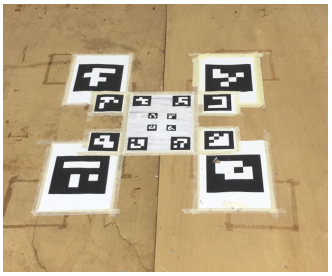

Fig. 17: ARUCO markers on the landing pad.

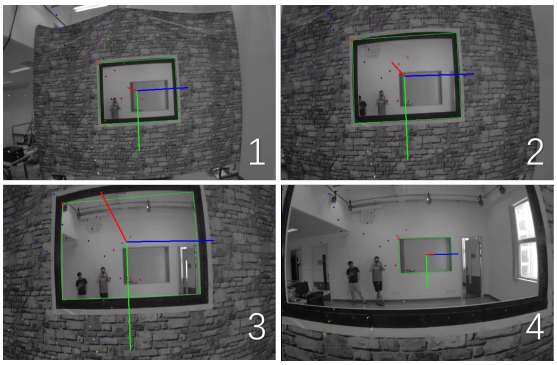

Fig. 18: Selected frames from the forward-looking camera.

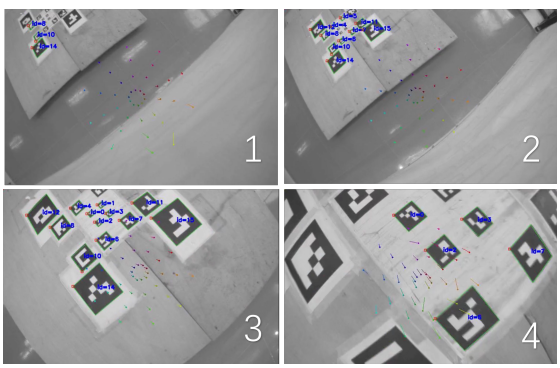

Fig. 19: Selected frames from the downward-looking camera.

window, and landing target. The motion capture system is able to accurately locate the position of the markers, from which ground truth position and orientation measurements are gathered. Note that, none of the measurements from the motion capture system are used in the proposed controller.

B. Experimental results

The experiments were conducted with the same control gains as the simulations. Before the proposed controller is triggered, the vehicle is hovering at position $\xi = [0.15, 1.79, -1.76]$ m, which is outside the space containing the landing pad. As mentioned in Section IV-C, there are four different modes during the full process of going through a window and landing on the target due to the limitation of the field of view of the on-board cameras. Fig. 18 shows the selected frames in a timed sequence from the forward-looking camera. These four frames are a fixed time step apart and were taken during a $mode$ 1 to $mode$ 2 transition. In Fig. 18 (1), (2), and (3), $mode$ 1 is active, and one can see that the window frame is well detected. In Fig. 18 (1), $t = T_1$ and the controller F_w is triggered. In Fig. 18 (2) and (3), the vehicle is still in $mode$ 1 and approaches the center of the window. As the vehicle approaches the window, the window frame disappears from the field of view of the camera and at time $t = T_2$ the $mode$ commutes to 2, as shown in Fig. 18 (4). Note that during transition from $mode$ 1 to 2, instead of losing the window frame, the camera may detect rectangles other then the target window, as depicted in Fig. 18 (4). In order to avoid this situation, the $mode$ changes from 1 to 2 if the pixel coordinates change instantaneously in a way that is incompatible with smooth tracking of the same window object. Fig. 19 shows the selected frames in a timed sequence from the downward-looking camera. These four frames were taken at fixed time steps during a transition from $mode$ 2 to 3 to 4. As shown in Fig. 19 (1), the vehicle has already crossed the window but the landing pad is not fully detected thus the $mode$ is still 2. At time instance $t = T_3$, as shown in Fig. 19 (2), the downward-looking camera detects successfully the landing pad, the $mode$ is switched to 3 and F_t is applied as control input. Recall that the switching from $mode$ 2 to $mode$ 3 is only triggered once in order to avoid inadequate behavior. In Fig. 19 (3), the vehicle approaches the target and the $mode$ is still 3. At the time instance $t = T_4$, when the quadrotor has almost reached the target position (see Fig. 19 (4)), the $mode$ changes to 4 and it is safe to slowly shutdown the motors.

Figures 20 and 21 show the position and velocity coordinates of the vehicle provided by VICON, respectively. We can see that the vehicle goes through the center of the window at the end of $mode$ 2 and finally lands on the target. Fig. 22 shows the evolution of the angular velocity and Fig. 23 show the evolution of the Euler angles. From Fig. 23, one can see that a good compromise in terms of time-scale separation between the outer-loop and inner-loop controllers is attained, which indicates that the inner-loop controller is sufficiently fast to track the outer-loop references, including during the transitions between different modes. Figure 25 shows the image feature $\bar{q}_w$ used for going through the window. The solid line represents $\bar{q}_w$ computed from the image sequence and the dashed line represents $\bar{q}_w$ provided by the VICON system. There are slight differences between these two computations due to the fact that rotation matrix R provided by the IMU is affected by the surrounding magnetic field generated by the fast rotating motors. Figures 27 and 26 show the translational optical flow used for going through the window and for landing respectively. The solid red lines represent the translational optical flow computed from the image sequence and the dashed line represents the translational optical flow derived from VICON measurements. The video of the experimental results can be found in https://youtu.be/j3UWk4qSIEE.

VII. Conclusion

This paper considers the problem of controlling a quadrotor to go through a window and land on planar target, using an Image-Based Visual Servo (IBVS) controller. For control

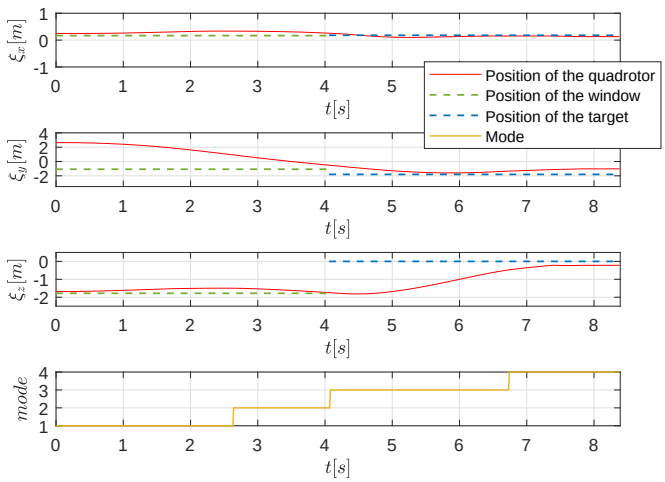

Fig. 20: Evolutions of the quadrotor's position ξ and the mode

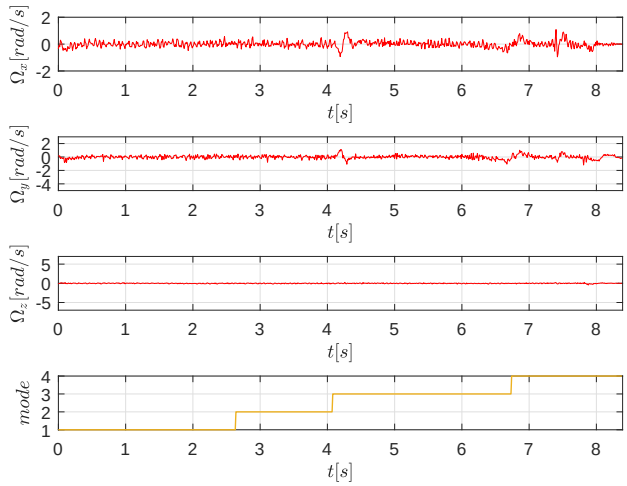

Fig. 22: Evolution of angular velocity Ω

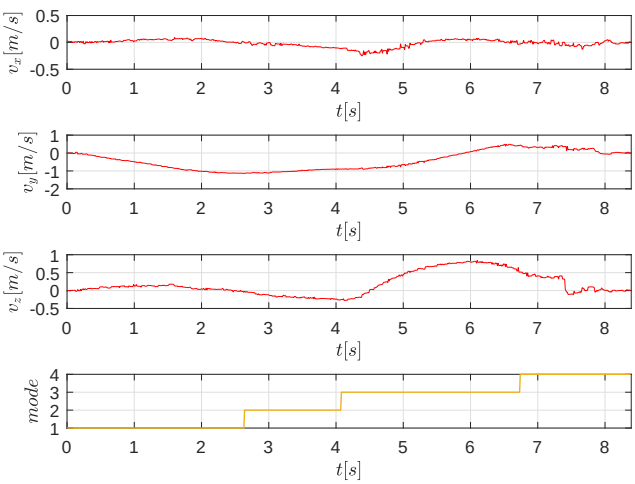

Fig. 21: Evolutions of the quadrotor's velocity v

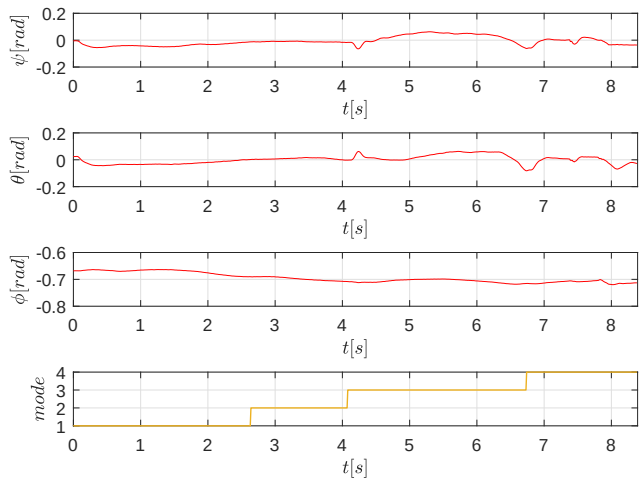

Fig. 23: Evolution of euler angle

purposes, the centroids vectors provided by the combination of the corresponding spherical image measurements of landmarks (corners) for both the window and the target are used as position feedback. The translational optical flow relative to the wall, window edges, and landing plane is used as velocity measurement. To achieve the proposed objective, no direct measurements of position or velocity are used and no explicit estimate of the height above the landing plane or of the distance to the wall is required. With the initial position outside the room containing the target, the proposed control law guarantees that the quadrotor aligns itself with the center line orthogonal to the window, crosses it with non-zero velocity and finally lands on the planar target successfully without colliding the wall or the edges of the window. The proof of convergence of the overall control scheme is provided and the simulation and experimental results show the effectiveness of the proposed controller.

Appendix

A. *Proof of Theorem 1*

Proof. The proof follows a reasoning very similar to that of Theorem 1 in Rosa et al. (2014). Recalling (1) and applying the control input (32), the closed-loop system can be written as

$$\begin{cases} \dot{\xi}_t = v \\ \dot{v} = -K_p^t q_t(\xi_t) - K_d^t \dfrac{v}{d_t}. \end{cases} \tag{40}$$

Before proceeding with the proof of item 1), a positive definite storage function $\mathcal{L}_2(\xi_t, v)$ will be defined and one will show that if $d_t(t)$ remains positive, $\dot{\mathcal{L}}_2$ is negative semi-

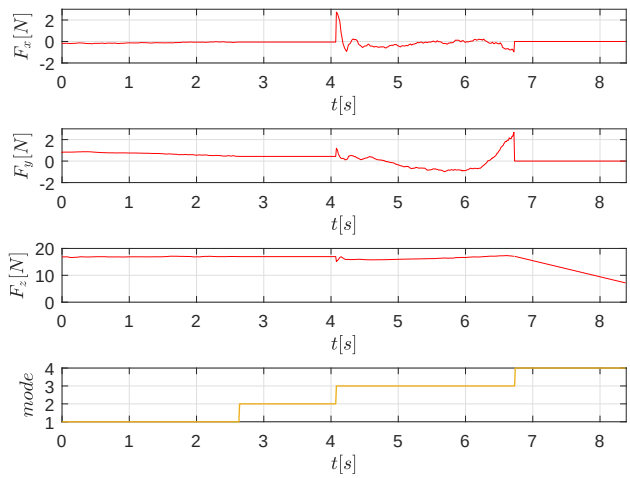

Fig. 24: Evolutions of the controller output F

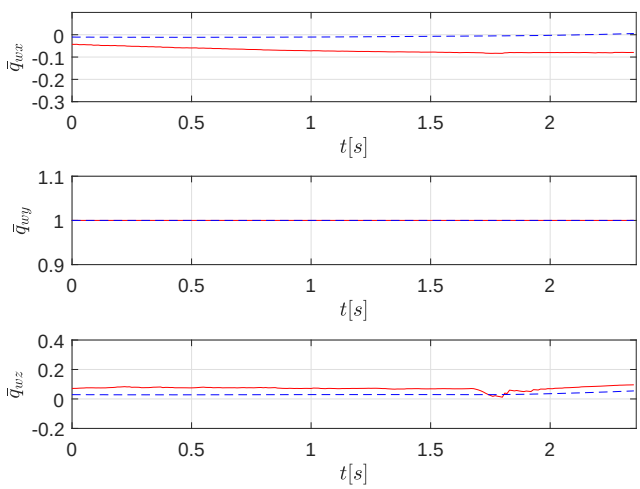

Fig. 25: Image features $\bar{q}_w$ computed from the image sequence (solid line) and from the VICON measurements (dashed line) during mode 1.

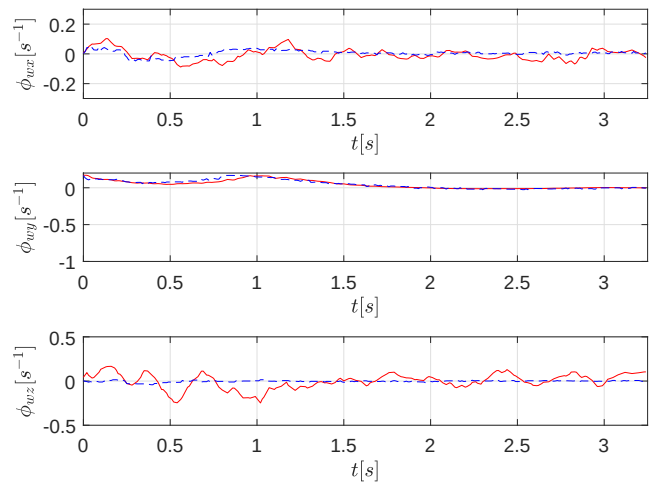

Fig. 26: Translational optical flow computed during $mode$ 1 (going through window) from the image sequence (solid line) and from the VICON measurements (dashed line).

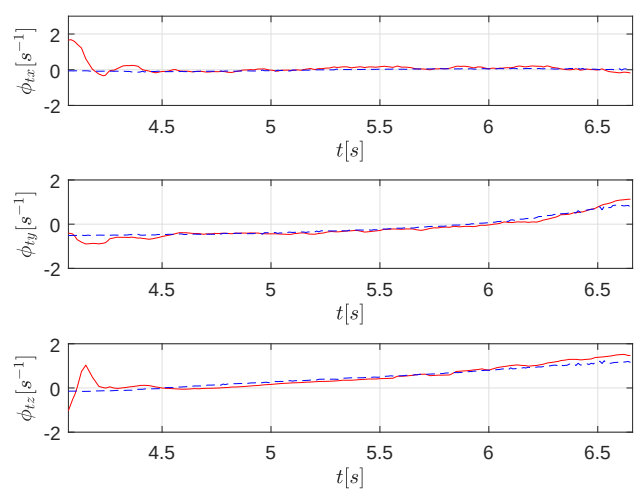

Fig. 27: Translational optical flow computed during $mode$ 3 (landing) from the image sequence (solid line) and from the VICON measurements (dashed line).

definite, which implies that the solutions remain bounded for all $t \geq 0$. Define $\mathcal{L}_2$ as

$$\mathcal{L}_2(\xi_t, v) = \mathcal{L}_1(\xi_t) + \frac{1}{2} v^\top K_p^{t^{-1}} v \tag{41}$$

where $\mathcal{L}_1(\xi_t)$ is the radially unbounded function given by

$$\mathcal{L}_1(\xi_t) = \frac{1}{n_t} \sum_{i=1}^{n_t} (\|P_i^t(\xi_t)\| - \|P_i^t(0)\|). \tag{42}$$

To show that $\mathcal{L}_1(\xi_t)$ is a positive definite function, note that

$$\frac{\partial \mathcal{L}_1}{\partial \xi_t} = q_t^\top \tag{43}$$

$$\frac{\partial^2 \mathcal{L}_1}{\partial \xi_t^2} = Q \tag{44}$$

where $Q = \frac{1}{n_t} \sum_{i=1}^{n_t} \frac{1}{\|P_i^t\|} \pi_{p_i^t}$ is positive definite, as long as at least two of the vectors p_i^t are non-collinear. It follows that $\mathcal{L}_1$ is a convex function of ξ_t, with a global minimum attained when $\frac{\partial \mathcal{L}_1}{\partial \xi_t} = q_t^\top = 0$, or equivalently, when $\xi_t = 0$. Since $\mathcal{L}_1(0) = 0$ is the global minimum of the function, one concludes that $\mathcal{L}_1(\xi_t)$ is positive-definite. Noting that,

$$\dot{\mathcal{L}}_1 = q_t^\top v, \tag{45}$$

it follows that

$$\dot{\mathcal{L}}_2 = -\frac{1}{d_t} v^\top K_p^{t^{-1}} K_d^t v \tag{46}$$

which is negative semi-definite as long as d_t remains positive and implies that the states $\xi_t(t)$ and $v(t)$ remain bounded for

all $t \geq 0$. The next steps of the proof consist in proving first Item (1) and then the uniform continuity of (46) along every system's solution in order to deduce, by application of Barbalat's Lemma, the asymptotic convergence of v to zero and from there one deduces the asymptotic convergence of $\dot{v}$ and then ξ_t to zero (Item 2).

Proof of Item 1: Using (40) and the fact that $d_t(t) = -\eta_t^\top \xi_t$ and $\dot{d}_t = -\eta_t{}^\top v$ yields

$$\ddot{d}_t = -k_{d_3}^t \frac{\dot{d}_t}{d_t} - k_{p_3}^t \beta_t \tag{47}$$

with

$$\beta_t(t) = -\eta_t^\top q_t = \tfrac{1}{n_t} \textstyle\sum_{i=1}^{n_t} \frac{d_t}{\|P_t^i\|} > 0, \ \forall t. \tag{48}$$

This relation is of course valid as long as $d_t(t) > 0$. From there, direct application of (Rosa et al. 2014, Th. 1-(2)) shows that if $d_t(0) \in \mathbb{R}^+$, the solution $(d_t, \dot{d}_t) \in (\mathbb{R}^+, \mathbb{R})$ exists and uniformly bounded $\forall t$ and converges asymptotically to $(0, 0)$.

Proof of Item 2: To show that

$$\ddot{\mathcal{L}}_2 = -\frac{2}{d_t} v^\top K_p^{t^{-1}} K_d^t \dot{v} + \frac{\dot{d}_t}{d_t} \frac{1}{d_t} v^\top K_p^{t^{-1}} K_d^t v$$

is bounded and hence $\dot{\mathcal{L}}_2$ (46) is uniformly continuous, it suffices to show that $\frac{\|v\|}{d_t}$ is bounded (so is $\frac{\dot{d}_t}{d_t}$). For that purposes, consider the dynamics of $\frac{v}{d_t}$:

$$\frac{d}{dt}(\frac{v}{d_t}) = -\frac{1}{d_t}((K_d^t + \dot{d}_t I)\frac{v}{d_t} + K_p^t q_t). \tag{49}$$

Since $\dot{d}_t$ converges asymptotically to zero and q_t is bounded then, by direct application of (Rosa et al. 2014, Lemma 4) one ensures that $\frac{v}{d_t}$ is bounded. From there one concludes that $\dot{\mathcal{L}}_2$ is uniformly continuous and hence v converges asymptotically to zero.

To prove that $q_t(t)$ (or equivalently ξ_t) is asymptotically converging to zero one has to show first $\dot{v}$ is converging to zero. From (40), one can verify that:

$$\ddot{v} = -\frac{K_d^t}{d_t}\dot{v} + \delta_{\dot{v}}^0, \tag{50}$$

with $\delta_{\dot{v}}^0 = K_d^t \frac{\dot{d}_t}{d_t} \frac{v}{d_t} - K_p^t \dot{q}_t$. Since $\frac{v}{d_t}$ (and hence $\frac{\dot{d}_t}{d_t}$) is bounded and

$$\dot{q}_t = Qv = Q_0 \frac{v}{d_t}, \text{ with } Q_0 = \frac{1}{n_t} \sum_{i=1}^{n_t} \frac{d_t}{\|P_i^t\|} \pi_{p_i^t} < I_3,$$

is also a bounded vector, one ensures that $\delta_{\dot{v}}^0$ is bounded. Therefore, direct application of (Rosa et al. 2014, Lem. 3) concludes boundedness and the asymptotic convergence of $\dot{v}$ to zero and hence one has:

$$\frac{v}{d_t} = -K_d^{t^{-1}} K_p^t q_t + o(t) \tag{51}$$

with $o(t)$ a asymptotically vanishing term.

By multiplying both sides of the above equation by the bounded vector $q_t^\top$ (the gradient of $\mathcal{L}_1$) and using the fact that $\dot{\mathcal{L}}_1 = q_t^\top v$ (45), one obtains:

$$\dot{\mathcal{L}}_1 = -d_t q_t^\top K_d^{t^{-1}} K_p^t q_t + d_t q_t^\top o(t). \tag{52}$$

Since $d_t(t)$ converges asymptotically to zero, then by taking the integral of (47)

$$\dot{d}_t(t) - \dot{d}_t(0) = -k_{d3}^t \log(\frac{d_t(t)}{d_t(0)}) - k_{p3}^t \int_0^t \beta_t(\tau) d\tau, \tag{53}$$

one concludes that

$$\lim_{t \to \infty} \int_0^t \beta_t(\tau) d\tau = +\infty. \tag{54}$$

Combining equation (54) with the fact that $d(t) \geq \beta_t(t)$ (from (48)) and replacing the time index t of equation (52) by the new time-scale index $s(t) := \int_0^t d_t(\tau) d\tau$ (s tends to infinity if and only if t tends to infinity), one has:

$$\frac{d}{ds}\mathcal{L}_1 = -q_t^\top K_d^{t^{-1}} K_p^t q_t + q_t^\top o(t), \tag{55}$$

from which one concludes that q_t (and ξ_t) is asymptotically converging to zero. □

B. Proof of Proposition 1

Proof. The proof follows and exploits the same technical steps of the proof of Theorem 1. Since assertions made are almost the same using either (33) or (32) (equivalently (33) with $\phi_t^* = 0$) except for the last item iii), the proof will be provided using (33) as feedback control and differences will be specified when necessaries.

When $\triangle \neq 0$ and $\phi_t^* \neq 0$, it is straightforward to verify that (46) becomes:

$$\dot{\mathcal{L}}_2 = -\frac{1}{d_t} v^\top K_p^{t^{-1}} K_d^t v + v^\top K_p^{t^{-1}} (\triangle + K_d^t \phi_t^* \eta_t) \tag{56}$$

Recall now the dynamics of $\dot{d}_t$ (47), $\frac{v}{d_t}$ (49), and of $\ddot{v}$ (50) in case where $\triangle \neq 0$ and $\phi_t^* \neq 0$.

$$\ddot{d}_t = -k_{d_3}^t \frac{\dot{d}_t}{d_t} - k_{p_3}^t \beta_t^\triangle \tag{57}$$

$$\frac{d}{dt}(\frac{v}{d_t}) = -\frac{1}{d_t}((K_d^t + \dot{d}_t I)\frac{v}{d_t} + \delta_v^\triangle) \tag{58}$$

$$\ddot{v} = -\frac{K_d^t}{d_t}\dot{v} + \delta_{\dot{v}}^\triangle, \tag{59}$$

with

$$\beta_t^\triangle(t) = \frac{1}{k_{p_3}^t}(k_{d_3}^t \phi_t^* + \eta_t^\top \triangle) - \eta_t^\top q_t \tag{60}$$

$$\delta_v^\triangle = K_p^t q_t - \triangle - K_d^t \eta_t \phi_t^* \tag{61}$$

$$\delta_{\dot{v}}^\triangle = K_d \frac{\dot{d}_t}{d_t} \frac{v}{d_t} - K_p^t \dot{q}_t + \dot{\triangle}. \tag{62}$$

Now since $\beta_t^\triangle(t) > 0, \forall t$ independently from the value chosen for ϕ_t^*, direct application of (Rosa et al. 2014, Th. 1-(2)) shows that the solution $(d_t, \dot{d}_t) \in (\mathbb{R}^+, \mathbb{R})$ exists and uniformly bounded $\forall t$ and converges (at least) asymptotically to $(0, 0)$.

By combining this with the fact that all terms involved in $\delta_v^\triangle$ (q_t, $\triangle$ and ϕ_t^*) are bounded, direct application of (Rosa et al. 2014, Lem. 4) concludes the boundedness of $\frac{v}{d_t}$. Since d_t is converging to zero, one concludes that v is converging to zero by a direct application of (Rosa et al. 2014, Lem. 3). Using the fact that $\dot{\triangle}$ is bounded by assumption, the proof

of boundedness $\ddot{v}$ (59) and its convergence to zero is directly deduced from to proof the unperturbed case (50). From there and analogously to the unperturbed case (Theorem 1- proof of Item 2), one gets:

$$\frac{v}{d_t} = -K_d^{t^{-1}} K_p^t q_t + K_d^{t^{-1}} \triangle + \eta_t \phi_t^* + o(t) \tag{63}$$

with $o(t)$ an asymptotically vanishing term.

By multiplying both sides of (63) by $q_t^\top$ and using the fact that $\dot{\mathcal{L}}_1 = q_t^\top v$ (45), one obtains:

$$\frac{\dot{\mathcal{L}}_1}{d_t} = -q_t^\top K_d^{t^{-1}} K_p^t q_t + q_t^\top (K_d^{t^{-1}} \triangle + \eta_t \phi_t^* + o(t)). \tag{64}$$

From there one distinguishes between the two issues stated in the proposition:

1) $\eta_t^\top \triangle(t) = 0, \forall t$ and $\phi_t^* = 0$ (F_t given by (32))

By changing the time scale index and similarly to argument used at the end of the proof of Theorem 1, one concludes that $\|q_t\|$ is ultimately bounded by $\frac{\|\triangle\|_{\max}}{k_{p_{1,2}}^t}$. Since $d_t = \eta_t^\top \xi_t$ converges to zero, one concludes that $\|\pi_{\eta_t} \xi_t\|$ is ultimately bounded by Δ_ξ which is the solution of $\|\pi_{\eta_t} q_t\| = \frac{\|\triangle\|_{\max}}{k_{d_{1,2}}^t}$.

2) $\eta_t^\top \triangle(t) \neq 0$, and $\phi_t^* \neq 0$ (F_t given by (33))

In that case one concludes that the storage function $\mathcal{L}_1$ is decreasing as long as the right hand side of the above equation is negative and $d_t > 0$ and hence ξ_t is bounded. The argument of changing the time index is not valid in this case. □

C. Proof of Proposition 2

Proof. We will consider hereafter only the case where $\sigma(q_w) = 1$ (or equivalently when $\eta_w^\top q_w < 0$). That is the situation in which the vehicle is going through the window while avoiding collision with the wall and the window edges.

From the dynamics of the closed-loop system (36), the proof focus first on the evolution of d_w. That is the evolution of the system in the direction η_w.

When $\|q_w(t)\| \geq \epsilon + \delta$, one has $d_w = d_o = -\eta_w^T \xi_w$ and hence:

$$\begin{aligned} \dot{d}_o &= -\eta_w^\top v \\ \ddot{d}_o &= -k_\phi^w \frac{\dot{d}_o}{d_o} - k_\phi^w \beta_w \end{aligned} \tag{65}$$

with $\beta_w = \phi_w^* + \frac{\eta_w^\top \triangle}{k_\phi^w} \geq \epsilon$.

When $\epsilon < \|q_w(t)\| < \epsilon + \delta$, one has $d_w = \frac{d_o d_e}{\alpha_w d_e + (1-\alpha_w) d_o}$ with α_w (defined by (19)) a uniformly continuous and bounded valued function on $[0, 1]$, and hence one verifies that:

$$\ddot{d}_o(t) = -k_\phi^w b(t) \frac{\dot{d}_o(t)}{d_o(t)} - k_\phi^w \beta_w \tag{66}$$

with $b(t) = \frac{(1-\alpha_w(t)) d_o(t) + \alpha_w(t) d_e(t)}{d_e(t)}$ a positive uniformly continuous and bounded function as long as $\epsilon < \|q_w(t)\| < \epsilon + \delta$ and $d_o(0) \in \mathbb{R}^+$. direct application of (Hérissé et al. 2012, Th. 5.1) to both equation (65) and (66), one can conclude that as long as $d_o(0) \in \mathbb{R}^+$ and $\|q_w(t)\| > \epsilon$ (or equivalently $\xi_w \notin \mathcal{W}$), $d_o(t) \in \mathbb{R}^+, \forall t \geq 0$ and $d_o(t)$ converges to zero exponentially (the exponential convergence of $d_o(t)$ is granted due to the fact that $\beta_w \geq \epsilon$) but never crosses zero and hence the vehicle will never touch the wall in a finite time. Additionally, one also proves, from (Hérissé et al. 2012, Th. 5.1), that there exists a finite time $t_1 \geq 0$ such that $\dot{d}_o(t) < 0, \ \forall t \geq t_1$ and hence d_o and d_w are decreasing after t_1.

When $\|q_w(t)\| \leq \epsilon$ (the situation when $\xi_w \in \mathcal{W}$), one has $d_w = d_e > d_o$. In this case one can easily verify that (65) can be rewritten as:

$$\ddot{d}_o = -k(t) \dot{d}_o - k_\phi^w \beta_w \tag{67}$$

with $k(t) = \frac{k_\phi^w}{d_e}$ a upper bounded positive gain as long as $d_e(t)$ is positive. Due to the fact that $\beta_w \geq \epsilon$, $\dot{d}_o$ is ultimately bounded by $-\frac{k_\phi^w \beta_w}{k(t)} \leq -\frac{k_\phi^w \epsilon}{k(t)}$ and hence one immediately ensures that there exists a finite time $t_2 \geq 0$ from which $\dot{d}_o(t) < 0, \forall t \geq t_2$. This implies that when $\|q_w(t)\| \leq \epsilon$ ($\xi_w \in \mathcal{W}$), d_o is decreasing $\forall t \geq t_2$ and hence d_o crosses zero in a finite time $\bar{t} > t_2$. Note that at $t = \bar{t}$, one has $\sigma(q_w(\bar{t})) = 0$ according to (35).

Consider now the the dynamics of the closed-loop system (36) in the plane π_{η_w}. That is the dynamics of $\xi_\perp := \pi_{\eta_w} \xi_w$. By defining $v_\perp := \pi_{\eta_w} v$ and $\triangle_\perp := \pi_{\eta_w} \triangle$, one gets:

$$\dot{\xi}_\perp = v_\perp \tag{68}$$

$$\dot{v}_\perp = -\frac{k_d^w}{d_w}(v_\perp + \frac{k_p^w}{k_d^w} \xi_\perp) + \triangle_\perp. \tag{69}$$

Define a new state

$$z = v_\perp + \frac{k_p^w}{k_d^w} \xi_\perp, \tag{70}$$

and the following positive definite storage function:

$$\mathcal{L}_3 = \frac{1}{2}\|z\|^2 + \frac{1}{2}(\frac{k_p^w}{k_d^w})^2 \|\xi_\perp\|^2,$$

with time derivative

$$\begin{aligned} \dot{\mathcal{L}}_3 &= -(\frac{k_p^w}{k_d^w})^3 \|\xi_\perp\|^2 - (\frac{k_d^w}{d_w} - \frac{k_p^w}{k_d^w})\|z\|^2 + z^\top \triangle_\perp \\ &\leq -(\frac{k_p^w}{k_d^w})^3 \|\xi_\perp\|^2 - \frac{\|z\|}{k_d^w}((\frac{k_d^{w\,2}}{d_w} - k_p^w)\|z\| - k_d^w \|\triangle_\perp\|), \end{aligned} \tag{71}$$

which is negative-definite provided that $0 < d_w < \frac{k_d^{w\,2}}{k_p^w}$ and $\|z\| \geq \frac{d_w k_d^w \|\triangle_\perp\|_{\max}}{k_d^{w\,2} - d_w k_p^w}$.

Proof of Item 1:

To show there exists a finite time $t_w \geq 0$ at which the vehicle enters the region $\mathcal{W}$ and remains there as long as $\sigma(q_w) = 1$, one proceeds using a proof by contradiction in two steps.

In the first step, assume that ξ_w is not converging to $\mathcal{W}$ in a finite time t_w and hence $\|q_w(t)\| > \epsilon$, $\forall t$. In the second one, assume that ξ_w is switching indefinitely between the two regions.

i) Consider the situation for which the initial condition is such that $\|q_w(0)\| > \epsilon$ (outside the region $\mathcal{W}$). Using the fact that there exists a finite time instant t_1 from which d_w is decreasing and converging to zero but never crosses zero

in finite time (see the above discussion), one concludes that z (70) is exponentially converging to zero and hence:

$$v_\perp = \dot{\xi}_\perp = -\frac{k_p^w}{k_d^w}\xi_\perp + o(t),$$

with $o(t)$ an exponential vanishing term. This in turn implies that $\xi_\perp$ (resp. $v_\perp$) is converging to zero exponentially. Combining this with the fact that $d_w(t)$ (resp. $d_o(t)$) is converging to zero, one concludes that there exists a finite time t_w at which $\|q_w(t_w)\| < \epsilon$ ($\xi_w(t_w) \in \mathcal{W}$), which contradicts the first part of the assumption.

ii) Consider the situation for which the vehicle is switching indefinitely between the two regions. Since $d_o(t)$ (respectively d_w) is decreasing $\forall t \geq \max\{t_1, t_2\}$ for both cases of $\|q_w\| > \epsilon$ and $\|q_w\| \leq \epsilon$ with the fact that $(\xi_\perp, v_\perp)$ converges exponentially to $(0, 0)$ (proof of the step (i)), one concludes that there exists a finite time $t_w \geq 0$ at which the vehicle enters the region $\mathcal{W}$ ($\|q_w(t_w)\| \leq \epsilon$), and remains there as long as $\sigma(q_w) = 1$, which contradicts the assumption.

Combining this with the discussion following (66), one ensures that there exists $\epsilon_1 > 0$ such that $d_w(t) \geq d_o(t) > \epsilon_1,\ \forall t < t_w$.

Proof of Item 2:

When ξ_w is inside the region ($\|q_w\| \leq \epsilon$), one guarantees that $\mathcal{L}_3$ (71) is decreasing as long as $0 < d_w < \frac{k_d^{w\,2}}{k_p^w}$ and $\|z\| \geq \frac{d_w k_d^w \|\triangle_\perp\|_{\max}}{k_d^{w\,2} - d_w k_p^w}$. Now since there exists a time $\bar{t} > t_w$ such that $d_o(\bar{t}) = 0$, one concludes that $t_{\lim}$ exists and it is equal to $\bar{t}$. □

References

Bertrand, S., Guénard, N., Hamel, T., Piet-Lahanier, H. & Eck, L. (2011), 'A hierarchical controller for miniature {VTOL} uavs: Design and stability analysis using singular perturbation theory', *Control Engineering Practice* **19**(10), 1099 – 1108.

Burton, A. & Radford, J. (1978), *Thinking in perspective: critical essays in the study of thought processes*, Vol. 646, Routledge.

Chaumette, F., Hutchinson, S. & Corke, P. (2016), Visual servoing, *in* 'Springer Handbook of Robotics', Springer, pp. 841–866.

DeSouza, G. N. & Kak, A. C. (2002), 'Vision for mobile robot navigation: A survey', *IEEE transactions on pattern analysis and machine intelligence* **24**(2), 237–267.

Falanga, D., Mueggler, E., Faessler, M. & Scaramuzza, D. (2017), Aggressive quadrotor flight through narrow gaps with onboard sensing and computing using active vision, *in* '2017 IEEE international conference on robotics and automation (ICRA)', IEEE, pp. 5774–5781.

Floreano, D. & Wood, R. J. (2015), 'Science, technology and the future of small autonomous drones', *Nature* **521**(7553), 460.

Guo, D. & Leang, K. K. (2020), 'Image-based estimation, planning, and control for high-speed flying through multiple openings', *The International Journal of Robotics Research* p. 0278364920921943.

Hamel, T. & Mahony, R. (2002), 'Visual servoing of an under-actuated dynamic rigid-body system: an image-based approach', *IEEE Transactions on Robotics and Automation* **18**(2), 187–198.

Hamel, T., Mahony, R., Lozano, R. & Ostrowski, J. (2002), 'Dynamic modelling and configuration stabilization for an x4-flyer.', *IFAC Proceedings Volumes* **35**(1), 217–222.

Hérissé, B., Hamel, T., Mahony, R. & Russotto, F.-X. (2012), 'Landing a vtol unmanned aerial vehicle on a moving platform using optical flow', *IEEE Transactions on Robotics* **28**(1), 77–89.

Ho, H., De Wagter, C., Remes, B. & De Croon, G. (2018), 'Optical-flow based self-supervised learning of obstacle appearance applied to mav landing', *Robotics and Autonomous Systems* **100**, 78–94.

Hutchinson, S., Hager, G. & Corke, P. (1996), 'A tutorial on visual servo control', *Robotics and Automation, IEEE Transactions on* **12**(5), 651–670.

Le Bras, F., Hamel, T., Mahony, R., Barat, C. & Thadasack, J. (2014), 'Approach maneuvers for autonomous landing using visual servo control', *IEEE Transactions on Aerospace and Electronic Systems* **50**(2), 1051–1065.

Loianno, G., Brunner, C., McGrath, G. & Kumar, V. (2017), 'Estimation, control, and planning for aggressive flight with a small quadrotor with a single camera and imu', *IEEE Robotics and Automation Letters* **2**(2), 404–411.

Mahony, R., Corke, P. & Hamel, T. (2008), 'Dynamic image-based visual servo control using centroid and optic flow features', *Journal of Dynamic Systems, Measurement, and Control* **130**(1), 011005.

Mahony, R. & Hamel, T. (2005), 'Image-based visual servo control of aerial robotic systems using linear image features', *IEEE Transactions on Robotics* **21**(2), 227–239.

Michael, N., Shen, S., Mohta, K., Mulgaonkar, Y., Kumar, V., Nagatani, K., Okada, Y., Kiribayashi, S., Otake, K., Yoshida, K. et al. (2012), 'Collaborative mapping of an earthquake-damaged building via ground and aerial robots', *Journal of Field Robotics* **29**(5), 832–841.

Peters, D. A. & HaQuang, N. (1988), 'Dynamic inflow for practical applications'.

Rosa, L., Hamel, T., Mahony, R. & Samson, C. (2014), 'Optical-flow based strategies for landing vtol uavs in cluttered environments', *IFAC Proceedings Volumes* **47**(3), 3176–3183.

Serra, P., Cunha, R., Hamel, T., Cabecinhas, D. & Silvestre, C. (2016), 'Landing of a quadrotor on a moving target using dynamic image-based visual servo control', *IEEE Transactions on Robotics* **32**(6), 1524–1535.

Serra, P., Cunha, R., Hamel, T., Silvestre, C. & Le Bras, F. (2015), 'Nonlinear image-based visual servo controller for the flare maneuver of fixed-wing aircraft using optical flow', *Control Systems Technology, IEEE Transactions on* **23**(2), 570–583.

Serres, J. R. & Ruffier, F. (2017), 'Optic flow-based collision-free strategies: From insects to robots', *Arthropod structure & development* **46**(5), 703–717.

Tang, Z., Cunha, R., Hamel, T. & Silvestre, C. (2018*a*), 'Aircraft landing using dynamic two-dimensional image-

based guidance control', *IEEE Transactions on Aerospace and Electronic Systems* **55**(5), 2104–2117.

Tang, Z., Cunha, R., Hamel, T. & Silvestre, C. (2018*b*), Going through a window and landing a quadrotor using optical flow, *in* '2018 European Control Conference (ECC)', IEEE, pp. 2917–2922.

Tang, Z., Li, L., Serra, P., Cabecinhas, D., Hamel, T., Cunha, R. & Silvestre, C. (2015), Homing on a moving dock for a quadrotor vehicle, *in* 'TENCON 2015-2015 IEEE Region 10 Conference', IEEE, pp. 1–6.

VICON (2014), 'Motion capture systems from vicon', http://www.vicon.com.

Zingg, S., Scaramuzza, D., Weiss, S. & Siegwart, R. (2010), Mav navigation through indoor corridors using optical flow, *in* 'IEEE International Conference on Robotics and Automation (ICRA)', pp. 3361–3368.