Three-dimensional position and velocity regulation of a quad-rotorcraft using optical flow — Source link 🔗

Luis Rodolfo Garcia Carrillo, Isabelle Fantoni, Eduardo Rondon, Alejandro Dzul

Institutions: Texas A&M University–Corpus Christi, University of Technology of Compiègne

Related papers:

- Landing a VTOL Unmanned Aerial Vehicle on a Moving Platform Using Optical Flow

- Vision Assisted Autonomous Landing of an Unmanned Aerial Vehicle.

- A new UAV configuration having eight rotors: Dynamical model and real-time control

- Adaptive Vision-Based Controller for Small Rotorcraft UAVs Control and Guidance

- Design and Evaluation of an Autonomous, Obstacle Avoiding, Flight Control System Using Simulated Visual Sensors

Three-dimensional Position and Velocity Regulation of a Quad-Rotorcraft Using Optical Flow

Luis Rodolfo Garcia Carrillo, Isabelle Fantoni, Eduardo Rondón, Alejandro
Dzul

3-dimensional Position and Velocity Regulation of a Quad-Rotorcraft Using Optical Flow

Luis Rodolfo García Carrillo, Isabelle Fantoni, Eduardo Rondón, and Alejandro Dzul

Abstract—We address the problem of hover flight and translational velocity regulation of a quad-rotorcraft UAV with the main objective of allowing the vehicle to navigate autonomously. This paper complements and improves previous researches considering multiple cameras systems, non conventional sensors, and which deal only with stabilizing the aerial vehicle in hover or in take-off and landing tasks. A vision system has been implemented to estimate the vehicle's altitude, lateral position, and forward velocity during flights. It is shown that, using visual information, it is possible to develop control strategies for different kinds of flying modes, such as hover flight and forward flight at constant velocity. The stability of the closed-loop system is ensured by implementing a hierarchical control strategy. In a first stage, the performance of the proposed methodologies was validated under a simulation environment, showing satisfactory and promising results. Next, real-time experimental applications, consisting of autonomous hover and forward flight at constant velocity were successfully achieved, validating the effectiveness of the proposed imaging algorithm and vision-based controller.

Index Terms—Quadrotor, quad-rotorcraft control, visual feedback, optical flow, hierarchical control, autonomous navigation.

I. INTRODUCTION

RESEARCH on Unmanned Aerial Vehicles (UAVs) continues to seek novel strategies to enable these kind of systems to perform with versatility. Common applications include assistance to soldiers, establishment of communication links, evaluation of risk in sensitive areas, weather, agriculture, and traffic monitoring, as well as search and rescue operations. Clearly, proficiency on sensing and control is fundamental to accomplish such missions successfully. Real-time stabilization with onboard sensors has been effectively demonstrated by several teams (e.g. [1] and [2]). Nevertheless, autonomous navigation motivates quite research, especially the studies that consider exploration of cluttered environments.

Imaging sensors are currently being used as an appealing ally to overcome situations previously considered as impassable. Cameras attached to the vehicle provide thorough information about the surrounding environment, but also furnish insight of the UAV motion. The interaction between sensor and environment enables the development of newfangled

L.R. García Carrillo is with the College of Science and Engineering, Texas A&M University - Corpus Christi, Corpus Christi, TX, USA. email: luis.garcia@tamucc.edu.

I. Fantoni and E. Rondón are with Heudiasyc, UMR CNRS 7253, Université de Technologie de Compiègne, Compiègne, FRANCE. email: {ifantoni,rondoned}@hds.utc.fr.

A. Dzul is with Instituto Tecnológico de la Laguna, Torreón, Coahuila, MEXICO. email: dzul@faraday.itlalaguna.edu.mx.

Fig. 1. Position stabilization and velocity regulation of a quad-rotorcraft during an autonomous flight. The proposed strategy makes use of a monocular imaging system and is based on optical flow computation.

techniques for navigation within disarrayed environments. For this reason, perception using vision algorithms has earned an important place as one of the essential components of every autonomous aerial vehicle.

Algorithms based on stereo vision have been employed in different kinds of missions and environments. The research addressed in [3] and [4] used stereo vision for controlling the height of a helicopter, as well as for developing a strategy for autonomous landing. In [5] a stereo system is combined with a monocular imaging sensor, enabling an aerial vehicle to navigate and avoid obstacles. An obstacle avoidance system for a quad rotorcraft based on stereo vision was presented in [6]. This method proved to be effective for creating a 3-dimensional map of the environment as well as for producing a collision-free path to a goal point. Similarly, a two-camera's system is presented in [7] to stabilize the position of a Vertical Take-Off and Landing (VTOL) vehicle, where each camera provides an image that allows controlling different dynamics of the aircraft. Monocular vision systems have also been implemented with encouraging results. Under this approach, the work addressed in [8] presented visual-servo control techniques for stabilizing the position of a quad-rotorcraft during autonomous flights.

Recently, bio-inspired techniques have reached the domain of UAVs development. In [9] an optical flow strategy inspired by bees' behavior was used for landing a fixed-wing UAV. Following a similar line of research, in [10] an optical flow-based controller was implemented to land a quad-rotorcraft autonomously. Terrain following applications based on optical

flow have also been addressed in [11], [12], and [13]. Inspired also by behaviors observed in nature, the authors in [14] developed an optical flow-based autopilot to regulate speed and enable lateral obstacles avoidance. A similar strategy was proposed in [15], where a 10-gram micro-UAV was built and equipped with sensors and controllers allowing obstacle avoidance using optical flow data.

While most of the previously mentioned methods consider vision-based tasks relying on multiple cameras systems [3]-[7], [14] and sometimes on non conventional sensors [11], [12], [15], our method explores the implementation of a monocular imaging system consisting of a conventional camera. In contrast to the solutions presented in [8], [9], [10], which deal with stabilizing the aerial vehicle in hover or in take-off and landing tasks, the research presented in this paper complements and improves these previous results by addressing the problem of stabilizing the 3-dimensional position and translational velocity of a quad-rotorcraft during autonomous flights. The research presented in the present article addresses the problem of stabilizing the 3-dimensional position and translational velocity of a quad-rotorcraft during autonomous flights. The proposed solution, which is based on a vision algorithm for line detection and optical flow computation, uses images furnished by a monocular camera system installed onboard the UAV. Such investigation involves two fundamental characteristics for any autonomous navigation system: we seek to accurately measure translational displacements, while eliminating the position drift when hovering. If the translational drift is correctly compensated, the hover flight can be used as an intermediary task between different flying behaviors, each one of them adapted to different conditions of the environment. Furthermore, velocity regulation is implemented to establish the different flying modes, such as lateral displacement and forward displacement actions. In order to make use of the optical flow in a very appropriate manner, a vision-based altitude controller has also been developed. The combination of these three vision-based controllers (hover flight, velocity regulation, and altitude stabilization) allows the vehicle to navigate autonomously over a road model in a real-time application. Two different kinds of missions were tested: position hold over a road segment and road following at constant velocity. A snapshot of the quad-rotorcraft while performing the latter of these tasks can be seen in Figure 1.

Preliminary studies and results of this work have been previously presented in two IROS conference articles [16], [17]. The research presented in this article complements and extends these previous works with substantial information about the quad-rotorcraft UAV platform and the vision system, as well as with additional information about the sensing and control algorithms proposed for navigating autonomously. Simulation studies are now being included, which serve for providing a better insight of the benefits, the limitations, as well as of the considerations that must be taken into account when implementing the proposed methods. The real-time application presented in [17] is now described with more details, and the experimental results obtained are presented in such a way that they are easier to interpret by the reader.

The body of this article is organized as follows. In Section II the considered application setup is described. Next, the image processing algorithms for estimating the vehicle's states are introduced in Section III. Section IV specifies the equations of the quad-rotorcraft dynamical model. The vision-based controller is obtained in Section V. After this, Simulation results are presented in Section VI, which where achieved by implementing the UAV dynamics, as well as the proposed imaging and control algorithms in a proprietary 3-dimensional synthetic environment. Section VII describes the experimental platform. The results obtained during autonomous flights are shown in Section VIII. Finally, the conclusions of this study as well as some current directions of our work are presented in Section IX.

II. VISION-BASED APPLICATION SETUP

The application setup consists of the quad-rotorcraft UAV platform equipped with an embedded camera pointing downward and the road model. This arrangement is shown in Figure 2 and can be summarized as follows:

- **Quad-Rotorcraft UAV:** The vehicle model considers 6 degrees of freedom (DOF). It has associated a body fixed frame $\mathcal{B} = (X_B, Y_B, Z_B)$, which is related to a fixed inertial frame $\mathcal{I} = (X_I, Y_I, Z_I)$. The inertial frame is located at the left corner of the beginning of the road, with the Z_I axis pointing upwards, the Y_I axis parallel to the road, and the X_I axis perpendicular and coplanar with Y_I. The body coordinate system is set in the center of the quad-rotorcraft with Z_B as the yaw axis, X_B as the pitch axis, and Y_B as the roll axis. The planes formed by (X_B, Y_B) and (X_I, Y_I) are considered parallel since the attitude stabilization keeps the pitch and roll angles near zero.
- **Pointing downward camera:** Since the camera is attached to the quad-rotorcraft, both of them are subject to the same movement. Considering a rigid scene, that is, only static objects in the camera's field of view (FOV), any sensed motion represents a displacement of the helicopter. The camera's coordinate system origin is located at the center of the camera, having the Z_C axis in the opposite direction with respect to Z_I and Z_B.
- **Road model:** The navigation mission considers a divided highway road, with a median separating both lanes. A model of the road is represented in Figures 2 and 3. The highway's median is modeled as a blue line of width K_b. Each border of the divided highway road is modeled by a red line of width K_r. The width of the road is constant and equal to Λ.

The imaging algorithm is in charge of computing an estimate of the vehicle's altitude, lateral position, and translational velocities. The relative altitude and lateral position information are deduced by extracting the road zone from the image, while the velocities are estimated using optical flow. All information sensed by the camera is related to the image plane.

III. IMAGE PROCESSING

Since optical flow computation requires knowledge of the distance between the camera and the sensed features, the
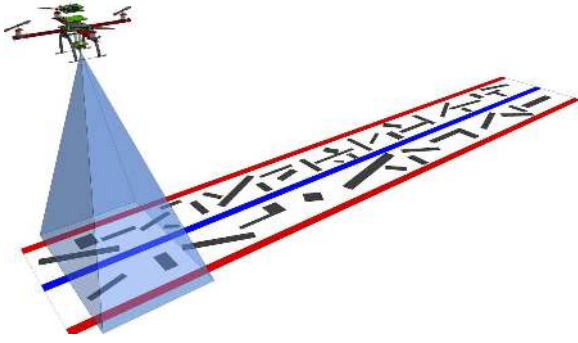
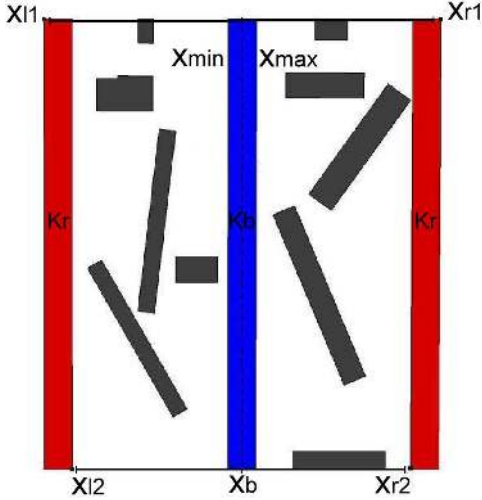Fig. 2. Quad-Rotorcraft navigation: a scheme of the proposed setup.

Fig. 3. The road model: From the group of highlighted image points the road zone can be effectively identified.

proposed imaging algorithm first estimates the helicopter's altitude. After this, the vision-based velocity control can be implemented. The complete development of this section has been proposed in [16] and [17], and a summary of the methodology is provided here.

A. Extracting the Road Zone

The first step to deduce the helicopter altitude consists of detecting the road zone, which is achieved by using a combination of color segmentation, a series of Gaussian filters, and line detection (Hough's Transform) algorithms which are applied to the road's image, see Figure 3. These results permit to estimate the points $x_{l_1}^i$, $x_{l_2}^i$, x_{min}^i, x_{max}^i, $x_{r_1}^i$ and $x_{r_2}^i$, where the subscript i corresponds to image-based coordinates. The computed points are, respectively, image projections of the world coordinates x_{l_1}, x_{l_2}, x_{min}, x_{max}, x_{r_1}, and x_{r_2}, which depict the left border, the central separation, and the right border of the road. From the group of extracted image points the road zone can be effectively identified.

B. Altitude and Lateral Position Estimation

The altitude estimation is obtained by means of the left and right carriageways of the road, where $x_{l_m}^i$ and $x_{r_m}^i$ represent their left and right border's centroids, respectively, in pixel coordinates

$$x_{l_m}^i = \frac{x_{l_2}^i + x_{l_1}^i}{2} \tag{1}$$

$$x_{r_m}^i = \frac{x_{r_2}^i + x_{r_1}^i}{2} \tag{2}$$

Let also α be defined as $\alpha = x_{r_m}^i - x_{l_m}^i$, which represents an image-based altitude variable. By considering that all points have the same depth z, one can write

$$x_{l_m}^i = \frac{f x_l}{z} \tag{3}$$

$$x_{r_m}^i = \frac{f x_r}{z} \tag{4}$$

where f represents the focal length of the camera. Introducing equations (3) and (4) into equations (1) and (2), one has

$$\alpha = \frac{f(x_r - x_l)}{z} = \frac{fW}{z} \tag{5}$$

where W is the distance between the left and right boarders, i.e. the width of the road. From this, it follows that

$$\frac{1}{\alpha} = K_{im} z \tag{6}$$

where K_{im} is a constant depending on the focal length of the camera and the width of the road, and z is the altitude of the quadrotor. From equation (6), it is possible to observe that controlling the inverse of the image-based altitude variable α is equivalent to stabilizing the altitude z of the helicopter.

The helicopter's lateral position can also be inferred from the extracted road points, specifically, from the lines passing through the points x_{min}^i, x_{max}^i, whose centroid x^i is the projection of the point x_b, see Figure 3. The objective here is to bring the image point x^i to the center of the image. By doing this, the vehicle's center of gravity, which is similar to that of the camera, will be aligned with respect to the center of the road model. Let's define ς as the image-based lateral position variable

$$\varsigma = \frac{f(x - x_b)}{z} \tag{7}$$

where x is the position of the helicopter's center of gravity in the inertial frame. A study of the time derivative is needed in order to measure the impact of an altitude variation on this variable. This procedure yields

$$\dot{\varsigma} = \frac{f}{z}\dot{x} - \frac{f(x - x_b)}{z^2}\dot{z} = \frac{f}{z}\dot{x} + \frac{(x - x_b)}{W}\dot{\alpha} \tag{8}$$

From equation (8) it is deduced that a previous stabilization of $\dot{\alpha}$ is a necessary requirement to fulfill a good lateral position stabilization. Once the altitude is maintained constant, the time derivative $\dot{\varsigma}$ will only depend of the lateral velocity of the helicopter, any other contribution will be considered as noise. Finally, when the altitude is stabilized $\dot{\alpha} = 0$, then, the time derivatives of the image-based variables becomes

$$\dot{\varsigma} = \frac{f}{z}\dot{x} \tag{9}$$

C. Translational velocities

The translational velocities and rotation rates of the camera are expressed in the inertial frame by (V_x, V_y, V_z) and (w_x, w_y, w_z), respectively. To accurately estimate the translational velocities of the vehicle, a tracking zone inside the road model is defined in such a way that the point $x^i = f\frac{x_b}{z}$ corresponds to the centroid of the zone. The algorithm of Shi and Tomasi [18] is used to identify the mesh of points that allows efficient tracking, i.e., the best textured points. Next, these features are used by the pyramidal implementation of the Lucas-Kanade algorithm [19] to accurately estimate the optical flow. Additionally, potential outliers are eliminated by verifying the standard deviation between the spatial mean of the translational optical flow and the optical flow in each tracked point.

The optical flow computed at point (x^i_k, y^i_k) is composed of a translational and a rotational part as (see [20] for details):

$$\begin{bmatrix} OF_{x^i_k} \\ OF_{y^i_k} \end{bmatrix} = T_{OF_k} + R_{OF_k} \tag{10}$$

with the translational part

$$T_{OF_k} = \frac{1}{z}\begin{bmatrix} -f & 0 & x^i_k \\ 0 & -f & y^i_k \end{bmatrix}\begin{bmatrix} V_x \\ V_y \\ V_z \end{bmatrix} \tag{11}$$

and the rotational part

$$R_{OF} = \begin{bmatrix} \frac{x^i_k y^i_k}{f} & -(f + \frac{(x^i_k)^2}{f}) & y^i_k \\ (f + \frac{(y^i_k)^2}{f}) & -\frac{x^i_k y^i_k}{f} & -x^i_k \end{bmatrix}\begin{bmatrix} \omega_x \\ \omega_y \\ \omega_z \end{bmatrix} \tag{12}$$

The spatial mean of the optical flow (mean of the optical flow on all tracked points) is then expressed as

$$\bar{OF}_x = \bar{V}_{OF_x} + K^x_x \bar{V}_{OF_z} + K^x_{xy}\omega_x - K^x_{xx}\omega_y + K^x_y\omega_z \tag{13}$$
$$\bar{OF}_y = \bar{V}_{OF_y} + K^y_y \bar{V}_{OF_z} + K^y_{yy}\omega_x - K^y_{xy}\omega_y - K^y_x\omega_z \tag{14}$$

with

$$\bar{V}_{OF_x} = -f\frac{V_x}{Z} \tag{15}$$
$$\bar{V}_{OF_y} = -f\frac{V_y}{Z} \tag{16}$$
$$\bar{V}_{OF_z} = \frac{V_z}{Z} \tag{17}$$

where $\bar{OF}_x$ and $\bar{OF}_y$ are the means of the sensed optical flow in the image coordinate system. $\bar{V}_{OF_x}$ and $\bar{V}_{OF_y}$ are the relative velocities of the vehicle in the camera coordinate system. $\bar{V}_{OF_z}$ is the inverse of the time-to-contact, known as the relative depth, and K^x_x and K^y_y are constant scale factors depending on the intrinsic parameters of the camera. These parameters can be computed separately and allow estimating the contribution of the helicopter's rotational movement to the optical flow. A discrete Kalman filter is then applied to compensate the rotational motion field of all tracked points as follows

$$\begin{bmatrix} \bar{V}_{OF} \\ W \end{bmatrix}_k = \begin{bmatrix} I & 0 \\ 0 & I \end{bmatrix}\begin{bmatrix} \bar{V}_{OF} \\ W \end{bmatrix}_{k-1} + \begin{bmatrix} \upsilon_1 \\ \upsilon_2 \end{bmatrix} \tag{18}$$

Here, $\bar{V}_{OF}$ is the vector composed of the mean translational components of the optical flow, W is the vector composed of angular velocities and υ_i represents the noise. This yields

$$\bar{V}_{OF} = [\bar{V}_{OF_x} + K^x_x \bar{V}_{OF_z}, \bar{V}_{OF_y} + K^y_y \bar{V}_{OF_z}]^T \tag{19}$$
$$= [\tilde{V}_{OF_x}, \tilde{V}_{OF_y}]^T$$

The measured outputs are the optical flow means $(\overline{OF}_x, \overline{OF}_y)$, and the angular velocity W_{IMU} provided by an embedded IMU. This terms are related to the state in equation (18) as

$$\begin{bmatrix} \overline{OF} \\ W_{IMU} \end{bmatrix}_k = \begin{bmatrix} I & K^T_R \\ 0 & I \end{bmatrix}\begin{bmatrix} \bar{V}_{OF} \\ W \end{bmatrix}_k + \begin{bmatrix} \nu_1 \\ \nu_2 \end{bmatrix}_k \tag{20}$$

where the noises υ and ν, in equation (18) and (20) respectively, are assumed to be uncorrelated noises with the following characteristics

- $\upsilon = [\upsilon_1, \upsilon_2]^T$, assumed to be a white noise with zero mean and known constant covariance matrix Q.
- $\nu = [\nu_1, \nu_2]^T$, assumed to be a white noise with zero mean and known constant covariance matrix R.

To compute the noise covariance matrix of the optical flow, the properties of the model are used. Since the quadrotor is modeled as a rigid body, the standard deviation between the spatial mean of the optical flow (e.g. $\tilde{V}_{OF_x}$) and the optical flow in each point (e.g. OF_{x_i}) is bounded. Any point having an optical flow too distant of the mean value will be ignored. The reliability of the optical flow is defined as a function of the desired constant bound σ^O_i, the total number of points having an estimate N_d, and the wrong measurements n_d. The noise covariance of the rotational measurements are considered constant and uncorrelated with the optical flow noise

$$R = \begin{bmatrix} f(\sigma^o_i, n_d, N_d) & 0 \\ 0 & R_{IMU} \end{bmatrix} \tag{21}$$

After the filter step, the three translational velocities can be estimated up to a scale factor by means of a simplified egomotion algorithm [21]. From the Kalman filter one has

$$\|OF_{x_i} - (K^x_{xy}\omega_x - K^x_{xx}\omega_y + K^x_y\omega_z)\| \leq \sigma^o_i \tilde{V}_{OF_x} \tag{22}$$
$$\|OF_{y_i} - (K^y_{yy}\omega_x - K^y_{xy}\omega_y - K^y_x\omega_z)\| \leq \sigma^o_i \tilde{V}_{OF_y} \tag{23}$$

The $N_d - n_d$ points satisfying (22) and (23) can be written as

$$\begin{bmatrix} OF_{x_i} \\ OF_{y_i} \end{bmatrix} = \bar{V}_{OF_z}\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}\begin{bmatrix} x_i \\ y_i \end{bmatrix} + \begin{bmatrix} \bar{V}_{OF_x} \\ \bar{V}_{OF_y} \end{bmatrix} \tag{24}$$

Here $\bar{V}_{OF_z}$ forms a virtual scaling factor c, the identity matrix forms a virtual rotation matrix R, and $\bar{V}_{OF_x}$ and $\bar{V}_{OF_y}$ form a translation vector t. The three translational components of the optical flow can be estimated by minimizing $\|OF_i - (cR\vec{x} + t)\|^2$ in a least square approach. Finally, a simplified ego-motion algorithm [21] allows to estimate the translational velocities up to a scale factor dictated by the camera's intrinsic and extrinsic properties. After all the previous steps have been performed, the translational velocities can be expressed as

$$\bar{V}_{OF_x} = -\frac{f\dot{x}}{z} = -\dot{\varsigma} \tag{25}$$
$$\bar{V}_{OF_y} = -\frac{f\dot{y}}{z} = -\dot{y}^i \tag{26}$$
$$\bar{V}_{OF_z} = \frac{\dot{z}}{z} \tag{27}$$

Fig. 4. Four rotor helicopter equipped with a camera.

where $(\dot{x}, \dot{y}, \dot{z})$ is the velocity vector of the vehicle's center of gravity. From the methodology previously explained, it is possible to verify that the vision algorithm provides the required information for performing altitude, lateral position, and forward velocity control of the quad-rotorcraft using only the image-based variables.

IV. DYNAMICAL MODEL OF THE AERIAL VEHICLE

The aerial vehicle used in the real time application is shown in Figure 4. The 3-dimensional position of the rotorcraft's center of gravity is expressed by $\xi = [x \, y \, z]^T \in \mathcal{I}$. In a similar way, the attitude of the vehicle is expressed by $\eta = [\phi \, \theta \, \psi]^T \in \mathcal{I}$, referred as roll, pitch and yaw, respectively. The corresponding equations for each one of the dynamics previously mentioned can be written as

$$m\ddot{x} = -u\sin\phi \tag{28}$$
$$m\ddot{y} = u\cos\phi\sin\theta \tag{29}$$
$$m\ddot{z} = u\cos\theta\cos\phi - mg \tag{30}$$
$$\ddot{\theta} = \tilde{\tau}_\theta \tag{31}$$
$$\ddot{\phi} = \tilde{\tau}_\phi \tag{32}$$
$$\ddot{\psi} = \tilde{\tau}_\psi \tag{33}$$

where $u, \tilde{\tau}_\theta, \tilde{\tau}_\phi, \tilde{\tau}_\psi$ are the collective thrust and the three torque inputs, respectively. The dynamic equations for the quad-rotorcraft model, show in (28)-(33), are simplified ones. When deriving (28)-(33), the following assumptions were considered: a) The dynamic model is obtained by representing the quad-rotorcraft as a solid body evolving in a 3-dimensional space, subjected to one force and three moments. b) The dynamics of the four electric motors are relatively fast, and therefore, they can be neglected. c) The non-rigidity of the rotor's blades is also neglected. d) The effect of wind affecting the performance of the aerial vehicle is ignored. e) The small body forces are ignored because they are generally of a much smaller magnitude than the principal control inputs force u and moments τ. For more details concerning the derivation of the simplified model the interested reader is referred to [22].

V. DEVELOPMENT OF THE VISION-BASED CONTROLLER

The quad-rotorcraft exhibit sub-actuated dynamics, therefore, the translational dynamics depends on the rotational dynamics. This Section provides details about the development of a vision-based control strategy, which considers proportional derivative (PD) controllers for stabilizing both subsystems. The strategies for stabilizing the vehicle's altitude (z) and heading angle (ψ) are introduced first.

A. Altitude and heading angle subsystems

Applying the vision-based altitude information in equation (30), the helicopter's altitude can be stabilized with the following control input

$$u = \frac{m(g - k_1^\alpha(\alpha^{-1} - \frac{1}{\alpha_{ref}}) - k_2^\alpha\dot{\alpha}^{-1})}{\cos\theta\cos\phi} \tag{34}$$

provided that $-\frac{\pi}{2} < \theta < \frac{\pi}{2}$ and $-\frac{\pi}{2} < \phi < \frac{\pi}{2}$, in such a way that the helicopter trajectory does not pass through any singularities [23]. In this control input, α^{-1} is the visual estimation of the altitude, and $\dot{\alpha}^{-1}$ is computed by numerical differentiation of the variable α^{-1}. Applying (34) into (30), and choosing the gains $k_1^\alpha = K_{im}k_p^z$ and $k_2^\alpha = K_{im}k_d^z$, the closed-loop dynamics can be stabilized by a simple PD control law. We consider the altitude error $z_e = z - z_{ref}$ in order to stabilize the vehicle at a certain reference z_{ref}, where the gains k_p^z and k_d^z are chosen to ensure that $s^2 + k_d^z s + k_p^z$ is a Hurwitz stable polynomial. If this assumption is valid, the altitude will converge to the reference value z_{ref}. Once the altitude converges, the longitudinal and the lateral subsystems can be treated as linear subsystems.

For the yaw dynamics case we obtain an independent system which consists of two integrators in cascade. The stabilization of the yaw dynamics is achieved by applying a control strategy of the form

$$\tilde{\tau}_\psi = -k_2^\psi\dot{\psi} - k_1^\psi\psi \tag{35}$$

The gains k_1^ψ and k_2^ψ are suitably chosen in such a way that $s^2 + k_2^\psi s + k_1^\psi$ is a Hurwitz stable polynomial. Therefore, the yaw dynamics is regularized to a value of zero.

B. Forward Displacement and Pitch Angle

The longitudinal subsystem focuses on the $(\dot{y}, \theta)$ subsystem stabilization. Using equation (34) in equation (29), leads to

$$m\ddot{y} = m(\nu_z + g)\tan\theta \tag{36}$$

where $\nu_z = -k_d^z\dot{z} - k_p^z(z - z_{ref})$. Given that the quadrotor is flying in hover, the pitch and roll angles are small enough in such a way that it is possible to consider $\tan\theta \approx \theta$. Also, since $z \to z_{ref}$ and $\nu_z \to 0$, the subsystem $(\dot{y}, \theta)$ can be rewritten as a linear subsystem

$$\ddot{y} = g\theta \tag{37}$$
$$\ddot{\theta} = \tilde{\tau}_\theta \tag{38}$$

Considering $y_1 = \frac{1}{g}(\dot{y} - V_y^{ref})$, the longitudinal velocity of the vehicle can de described by the following chain of integrators:

$$\dot{y}_1 = y_2 \tag{39}$$
$$\dot{y}_2 = y_3 \tag{40}$$
$$\dot{y}_3 = \tilde{\tau}_\theta \tag{41}$$

where y_1 is the error between the desired and the real velocity, $y_2 = \theta$ and $y_3 = \dot{\theta}$. The hierarchical control is constructed by separating, in the previous subsystem, the rotational dynamics from the translational dynamics (low-level control and navigation control respectively). Considering y_2 as the control input for the translational dynamics, and applying a backstepping change of variables, one has

$$\dot{y}_1 = y_2^{ref} + \tilde{y}_2 \tag{42}$$
$$\dot{\tilde{y}}_2 = \tilde{y}_3 \tag{43}$$
$$\dot{\tilde{y}}_3 = \ddot{y}_2^{ref} + \tilde{\tau}_\theta \tag{44}$$

where $y_2^{ref} = \theta_{ref}$, $\tilde{y}_2 = \theta - \theta_{ref}$, and $\tilde{y}_3 = \dot{\theta} - \dot{\theta}_{ref}$. θ_{ref} represents the reference angle that will be computed using the mean value of the optical flow direction. Since the attitude controller works at a higher rate than the translational dynamics controller, the dynamics of the reference angle can be ignored ($\ddot{\theta}_{ref} = \dot{\theta}_{ref} = 0$). In fact, the pitch dynamics will converge faster than the longitudinal dynamics. Then, one has

$$\dot{y}_1 = y_2^{ref} + \tilde{y}_2 \tag{45}$$
$$\dot{\tilde{y}}_2 = \tilde{y}_3 \tag{46}$$
$$\dot{\tilde{y}}_3 = \tilde{\tau}_\theta \tag{47}$$

Choosing $y_2^{ref} = -k_{OF_y}(\overline{V}_{OF_y} - \overline{V}_{OF_y}^{ref})$, where k_{OF_y} is a known constant depending on the camera's calibration parameters, leads to the equivalent notation $y_2^{ref} = -k_y^p y_1$. Then, the control signal is finally given by

$$\tilde{\tau}_\theta = -k_d^\theta \dot{\theta} - k_p^\theta(\theta - \theta_{ref}) \tag{48}$$

The closed-loop system is then represented by $s^3 + k_d^\theta s^2 + k_p^\theta s + k_p^\theta k_p^y$, which is a Hurwitz stable polynomial if $k_d^\theta, k_p^\theta, k_p^y > 0$ and $k_d^\theta > k_p^y$. Therefore, the dynamics $\dot{y} - \theta$ are stabilized according to the desired references.

C. Lateral Position and Roll Angle

Considering that the vehicle's altitude has reached the desired reference, the (x, ϕ) subsystem can be expressed as a linear subsystem composed of a chain of integrators

$$\ddot{x} = -g\phi \tag{49}$$
$$\ddot{\phi} = \tilde{\tau}_\phi \tag{50}$$

The vehicle's lateral dynamics can be described using the following chain of integrators

$$\dot{x}_1 = x_2 \tag{51}$$
$$\dot{x}_2 = x_3 \tag{52}$$
$$\dot{x}_3 = x_4 \tag{53}$$
$$\dot{x}_4 = \tilde{\tau}_\phi \tag{54}$$

Let's consider ϕ as the control input for the lateral translational dynamics, then, the chain of four integrators can be separated into two subsystems composed of two integrators for both of them. The implementation of a backstepping change of

variables, gives

$$\dot{x}_1 = x_2 \tag{55}$$
$$\dot{x}_2 = x_3^{ref} + \tilde{x}_3 \tag{56}$$
$$\dot{\tilde{x}}_3 = x_4 \tag{57}$$
$$\dot{x}_4 = \tilde{\tau}_\phi \tag{58}$$

where $\tilde{x}_3 = \phi - \phi_{ref}$, and $x_3^{ref} = \phi_{ref}$. This reference angle is computed by using $(\varsigma, \dot{\varsigma})$ from equations (7) and (25) respectively. The reference angle dynamics are ignored since the rotational dynamics of the vehicle are faster than the translational dynamics and also because the low level autopilot was designed using high gains. The image-based variables $(\varsigma, \dot{\varsigma})$ are used to stabilize the position, as follows

$$x_3^{ref} = \phi_{ref} = -k_2^x \dot{\varsigma} - k_1^x \varsigma \tag{59}$$
$$\phi_{ref} = -k_d^x \dot{x} - k_p^x(x - x_b) \tag{60}$$

and

$$\tilde{\tau}_\phi = -k_d^\phi x_4 - k_p^\phi(\tilde{x}_3) = -k_d^\phi \dot{\phi} - k_p^\phi(\phi - \phi_{ref}) \tag{61}$$

The closed-loop system is stable if $k_p^\phi, k_d^\phi, k_p^x, k_d^x > 0$, $k_d^\phi > k_d^x$, and $k_p^\phi > \frac{(k_d^\phi)^2 k_p^x}{k_d^x(k_d^\phi - k_d^x)}$. By doing this, the roll angle, the lateral position, as well as the lateral velocity reach their desired values.

VI. SIMULATION RESULTS

The effectiveness of the vision-based controller is studied first in a simulation experiment. The main idea is to validate the proposed methodology under realistic conditions before implementing it in the real-time application. Aiming at this goal, a proprietary dynamical simulator was used [24], which consists of a 3-dimensional synthetic environment where the quad-rotorcraft can fly manually or autonomously. This simulator is capable of computing all the vehicle states since the dynamical model equations (28)-(33) are taken into account. In addition, the simulator gives us the possibility of considering virtual cameras attached to the quad-rotorcraft. Specifically, it creates synthetic images of what the cameras perceive while the vehicle moves across the environment. Those synthetic images can be used as if they were the real images received from a vision system. Therefore, the imaging algorithm and control law that will be implemented with the real system can be used directly in this simulation tool for processing the synthetic images and computing the required control inputs.

A. 3-dimensional synthetic environment

Figure 5 shows a snapshot of the quad-rotorcraft simulator environment, which consists of three windows. The main window (located to the right) is divided in four sections. The upper section of this window shows the quad-rotorcraft in the 3-dimensional environment as perceived from an external beholder. The lower section of the frame is divided in three frames. The frame on the left corresponds to the synthetic view of a pointing downwards camera, while the frame on the right corresponds to a camera pointing to the front of the vehicle. The black section between these two synthetic views
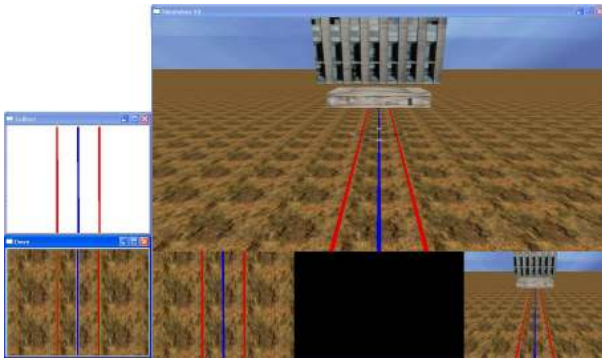
Fig. 5. Snapshot of the quad-rotorcraft simulator environment. The window in the right hand side is divided in four sections. The upper section shows the quad-rotorcraft in the 3-dimensional environment as perceived from an external beholder. The lower frame on the left corresponds to the synthetic view of a pointing downwards camera, while the frame on the right corresponds to a camera pointing to the front of the vehicle. The black section between these two synthetic views will be used for further developments. The two small windows located to the left of the main window show to the image processing task. The small lower window represents the camera's image that is being used for computing information of interest. Finally, the small upper window shows the output of the image processing algorithm.

is currently empty (black image), but it will be used for further developments. The two small windows located to the left of the main window are devoted to the image processing task. The small lower window represents the camera's image that is being used for computing information of interest (in this case, the pointing downwards camera). Finally, the small upper window shows the output of the image processing algorithm. The computer vision algorithm running in the synthetic image is based on OpenCV functions [25].

B. A study of the altitude dynamics estimation algorithm

The performance achieved by the algorithm for estimating the platform's altitude is studied first. This is done by processing the synthetic image provided by the pointing downwards camera. The test protocol performed is as follows. First, a manual take-off is performed, followed by an autonomous ascension stage which lasts until a maximum altitude of 20 meters is reached. After this, the altitude is decreased gradually and the test ends with a manual landing. This experiment is performed with the lateral and longitudinal position well stabilized in a previously defined setpoint. The outcome of this test is shown in Figure 6. In this figure, the real altitude as computed by the simulator's physical model is traced in red, while the altitude estimated by the vision algorithm is traced in blue.

This experiment demonstrated that the vision algorithm is effective for estimating the variable $\frac{1}{K_{im}\alpha}$. It can be deduced that, as the quad-rotorcraft goes higher, the width of the road zone in the image becomes smaller. However, since the inverse of the width of the road is implemented in our computations, the resulting variable follows the same behavior than the real altitude of the vehicle. Despite the good performance of the vision-based altitude estimation, there are some issues in the implementation of this method that must be discussed
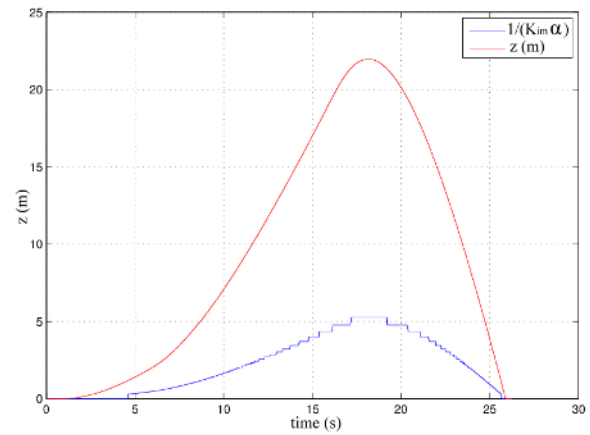

Fig. 6. The performance of the algorithm for estimating the platform altitude. First, a manual take-off is performed, followed by an autonomous ascension stage which lasts until a maximum altitude of 20 meters is reached. After this, the altitude is decreased gradually and the test ends with a manual landing. The real altitude as computed by the simulator's physical model is traced in red, while the altitude estimated by the vision algorithm is traced in blue.

here, specifically, the existence of two discontinuities in the estimation process. The first discontinuity is encountered when the road zone first appears in the camera's field of view. At this moment, the algorithm starts to estimate the altitude and the variable passes directly from zero to the estimated value. On the other hand, when the road zone comes out of the camera field of view, it is not possible for the algorithm to estimate the vehicle's altitude. In such situation the variable goes directly to zero creating the second discontinuity. Since those discontinuities happen in low altitudes (near the ground level), it can be deduced that this issue will mostly affect the take-off and the landing tasks. However, take-off and landing can be effectuated in a open-loop form, relying only on the velocities computed by the optical flow algorithm for avoiding translational drift.

C. Simulations results under autonomous flights

A study of the performance obtained with the proposed navigation controller is now presented. The objective of this simulation consists of regulating the forward velocity at 1 m/s by means of the computed optical flow, a process that has to be done while maintaining a relative lateral position around the origin (center of the road) and while flying at a constant altitude of 5 meters. The behavior of the quad-rotorcraft attitude dynamics during the experiment is shown in Figure 7. Similarly, the vehicle's translational velocities are shown in Figure 8. Finally the corresponding 3-dimensional position is shown in Figure 9. Figure 7 shows that during the first instants of the simulation, the attitude angles of the platform converge to specific values very close to zero. This is due to the fact that the attitude controllers are operating from the beginning of the test. Then, the position controllers are activated at around 27 seconds of time. At this moment it can be observed that, for a constant forward velocity, the pitch angle converges to a desired reference angle. Notice also that the pitch angle overruns the reference angle (1 degree) while the vehicle is in the first instants of the forward displacement. Moreover, it can
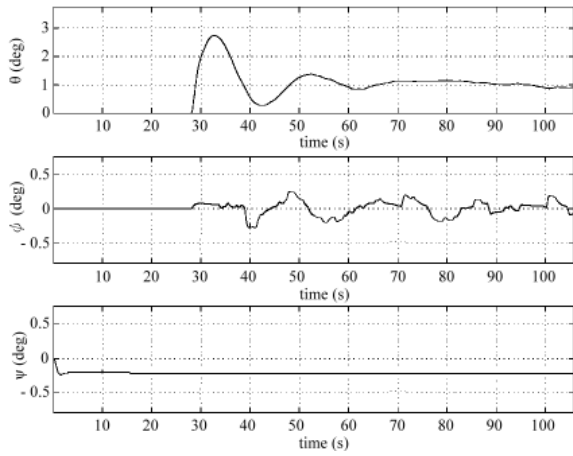
Fig. 7.   Simulation results: behavior of the quad-rotorcraft attitude dynamics while performing the navigation experiment. Notice that the pitch angle θ is stabilized at around 1 degree, which ensures a constant forward velocity. The roll angle ϕ is controlled for stabilizing the lateral position above the center of the road. The yaw angle ψ is regulated at a value which is very close to zero with a small steady state error of about 0.25 degrees. This small error is mainly due to the fact that a PD controller is being implemented for stabilizing the heading dynamics.
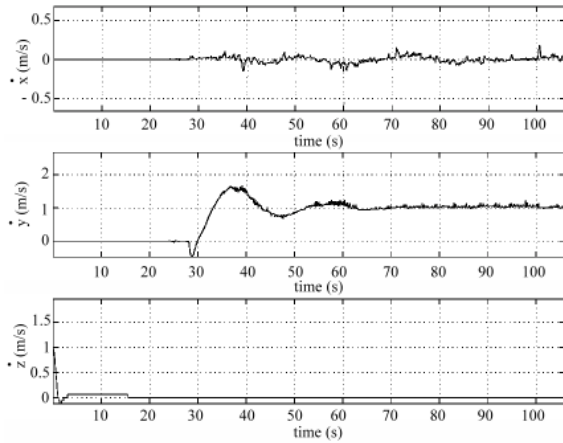
Fig. 8.   Simulation results: behavior of the quad-rotorcraft translational velocities while performing the navigation experiment. The forward velocity $\dot{y}$ is stabilized at around 1 m/s. The lateral velocity and altitude velocity ($\dot{x}$ and $\dot{z}$), respectively, are maintained around zero.
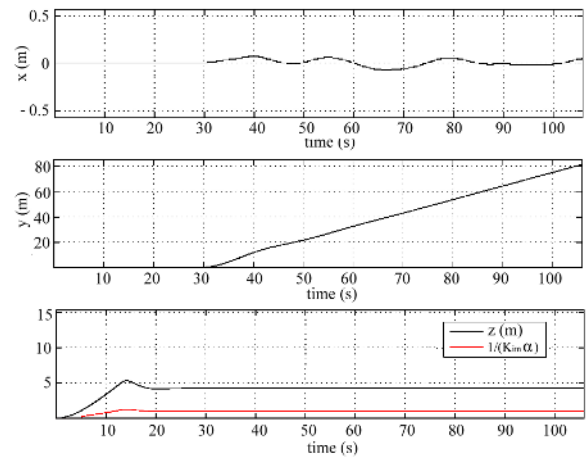

Fig. 9.   Simulation results: behavior of the quad-rotorcraft translational position while performing the navigation experiment. From the middle graphic, it can be seen that the quad-rotorcraft performs a displacement of around 80 meters during 70 seconds of flight. The lateral position stabilization, shown in the upper graphic, presents a good performance also, maintaining the vehicle around the central section of the road. From the lower graphic, notice that the quad-rotorcraft altitude z is stabilized at around 5 meters, and that the visual variable $\frac{1}{K_{im}\alpha}$ follows exactly the same behavior.

be seen that the proposed control does not create oscillations in the pitch and roll dynamics.

From Figure 8, it can be deduced that the proposed vision-based controller maintains the quad-rotorcraft forward velocity ($\dot{y}$) to a constant value of 1 m/s, while stabilizing the lateral velocity to a value around zero. Furthermore, the instantaneous velocity remains constant without any discontinuities that could be caused when the vision algorithm recomputes the rich textured features required for estimating the optical flow.

Finally, Figure 9 shows the evolution of the quad-rotorcraft translational position obtained from the simulator during the experiment. From the middle graphic, it can be seen that the quad-rotorcraft performs a displacement of around 80 meters during 70 seconds of flight. The lateral position stabilization, shown in the upper graphic, presents a good performance also, maintaining the vehicle around the central section of the

road. From the lower graphic, notice that the quad-rotorcraft altitude (z) is stabilized at around 5 meters, and that the visual variable $\frac{1}{K_{im}\alpha}$ follows exactly the same behavior.

These simulation results allow one to conclude that the proposed strategy is effective for controlling the quad-rotorcraft during autonomous flights. Thus, the study can continue with the experimental implementation of the vision-based controller.

VII. EXPERIMENTAL PLATFORM

The experimental platform used for the present studies is shown in Figure 10. It consists of a quad-rotorcraft aerial vehicle, a supervisory ground station PC, and an onboard vision system. The communication between the quad-rotorcraft and the ground station is ensured by means of a wireless data link. A block diagram showing this system architecture is presented in Figure 11.

A. Quad-Rotorcraft Aerial Vehicle

The quad-rotorcraft helicopter shown in Figure 4 was built from a group of off-the-shelf components. The vehicle is equipped with brushless motors and has a total weight of approximately 750 grams. Same axis rotors are separated by a distance of 40 cm. Power is provided by a 12 V, 2200 mAh Li-Po battery, which enables an autonomy of about 15 minutes. The onboard electronics consists of two interconnected cards, see Figure 12. The first board contains the control unit, while the second one considers the motor's speed controllers. The control unit card performs the essential tasks of sensing, communicating, and computing the control law for stabilizing the UAV attitude during flight. The characteristics of this board can be summarized as follows:

- Texas Instruments® **TMS320F2812 DSP** module. It reads the signals of the embedded sensors and computes
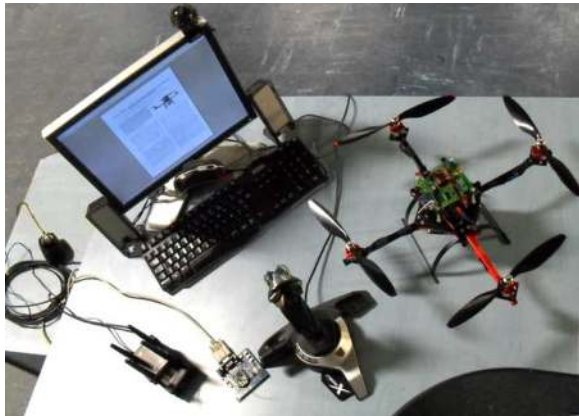
Fig. 10. The experimental platform used for the present studies consists of a quad-rotorcraft aerial vehicle, a supervisory ground station PC, and an onboard vision system. The communication between the quad-rotorcraft and the ground station is ensured by a wireless data link.
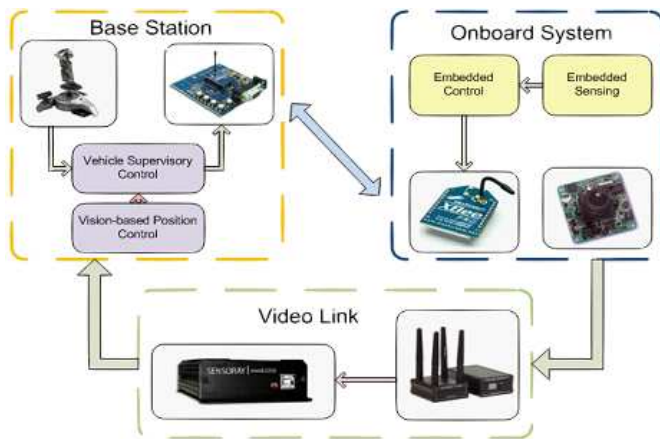

Fig. 11. Block diagram showing the experimental system architecture.

the control law for stabilizing the aircraft. Its working frequency is 500 Hz.

- **MIDG II INS/GPS IMU** from Microbotics Inc®. It measures the angular position of the quad-rotorcraft at a frequency of 100 Hz.
- Three **ADXRS150 analog gyroscopes**. They measure the angular rates at 500 Hz. We have chosen analog rate measurement rather than IMU based measurements since we can obtain a faster refresh of angular rates, which enables a better attitude stabilization of the UAV.
- **Battery voltage measurement circuit**. It provides the actual tension level of the power supply battery. This information is used for performing a safe landing before an unwanted discharge of tension occur (avoiding accidents).
- XBee ZB ZigBee PRO® **Radio Modem**. It is used to communicate the ground station and the aerial vehicle. This wireless link can be used to introduce external control inputs, send the sensors' information to the ground station, etc.

Furthermore, the second board contains:

- **Signal conditioning circuitry**. In this board, the motor's control signals are decoupled from the rest of the electronic systems. PWM signals are also filtered and

Fig. 12. The electronics on board: (a) Electronic card for the DSP, analog rate gyros, IMU connections and wireless modem; (b) Signal conditioning board.
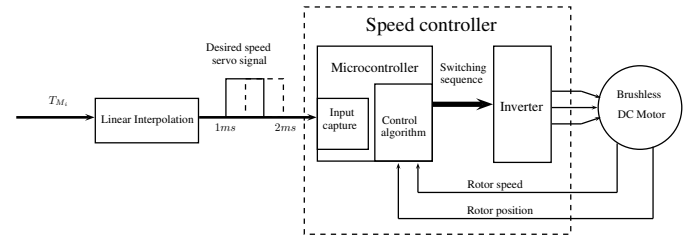

Fig. 13. A block diagram describing how the control signals reach the quad-rotorcraft motors.

conditioned. A block diagram describing how the control signals reach the quad-rotorcraft motors is shown in Figure 13. More specific details can be found in [26].

B. Supervisory ground station

The supervisory ground station is composed of a desktop PC, a flight simulator Cyborg-X® joystick and an XBee ZB ZigBee PRO Radio Modem, which are shown in Figure 14. In addition, a Diversity video system receiver, which can be seen in Figure 15 (b), allows receiving real-time video from the quad-rotorcraft onboard camera. The ground station executes a supervisory control application allowing a user to send information to the aerial vehicle and to chose between a manual control or an autonomous vision-based flight. The supervisory ground station receives and saves data sent by the vehicle in order to debug and analyze the flight experiments. Image processing (extraction of the road zone and optical flow computation) is carried out in the ground station PC. The control feedback between the supervisory ground station and the quad-rotorcraft is performed at 30 Hz.

C. Vision system

The UAV vision system components are shown in Figure 15. Real-time video is obtained by means of a high definition CTDM-5351® camera. It has a resolution of 640×480 pixels and is installed (pointing downwards) in the lower part of the helicopter. The camera is connected to a 200 mW video and audio HF transmitter. Images from the vision system are recovered on ground by a 4-Antenna Diversity® System Receiver which is connected to the supervisory ground station PC throughout a USB frame grabber. The frequency of the video transmission is performed at a rate of 30 Hz. Once
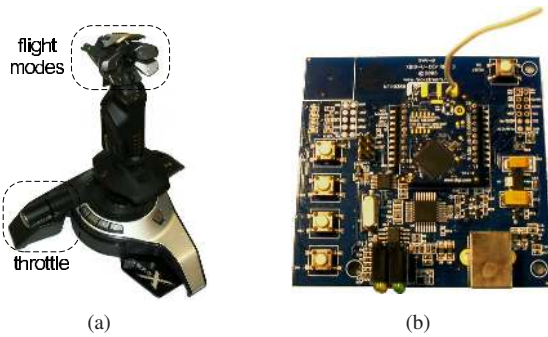
(a)   (b)

Fig. 14. Supervisory ground station components: (a) The Flight simulator Cyborg-X® joystick can be used for enabling the autonomous flight mode, or for taking manual control of the vehicle, if needed; (b) An XBee ZB ZigBee PRO Radio Modem allows sharing information between the aerial vehicle and the ground station PC.

(a)   (b)

Fig. 15. The UAV vision system: (a) CTDM-5351 camera installed on board, in the lower part of the quad-rotorcraft; (b) 4-Antenna Diversity System Receiver connected to the ground station PC.

received by the supervisory ground station, real-time video is processed by the computer vision application developed for computing the 3-dimensional position and the translational velocity of the quad-rotorcraft. The computer vision algorithms are programmed in Visual C++ and are based on OpenCV functions [25].

VIII. REAL-TIME EXPERIMENTS

In order to validate the vision-based control strategy we have performed two main experiments: hover flight and forward flight at constant velocity. The hover flight stabilization experiment was executed over a specified zone of the road model. As mentioned previously, in order to estimate the variable $\frac{1}{\alpha}$ the total width of the road model needs to be under the field of view of the camera. For this reason, the control strategy is activated once the vehicle is already flying at a desired altitude. Figure 16 shows an image from the road environment taken by the vehicle's camera during the real-time experiment. The Figures 17, 18, and 19 show the behavior of the vehicle's states (position, linear velocities and Euler angles) during the experiment. Figure 17 shows that the translational displacement is effectively eliminated, and that the altitude is well stabilized. It can be observed in Figure 19 that the Euler angles dynamics are not affected by the controller acting on the position stabilization. A video of the quad-rotorcraft while performing such an experiment can be seen at

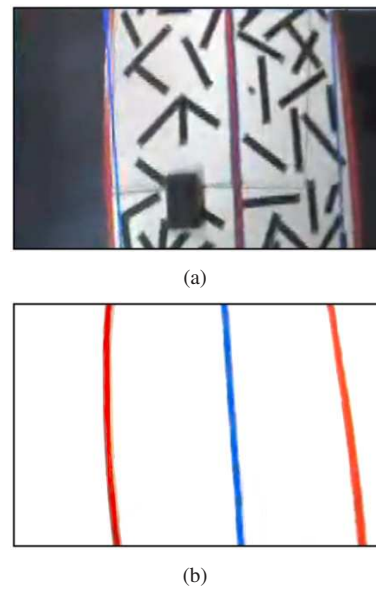(http://youtu.be/xPb-IHSsNIo).

(a)

(b)

Fig. 16. Image of the road environment taken by the vehicle's camera during the experiment: (a) Raw image as provided by camera; (b) Resulting image after image processing.
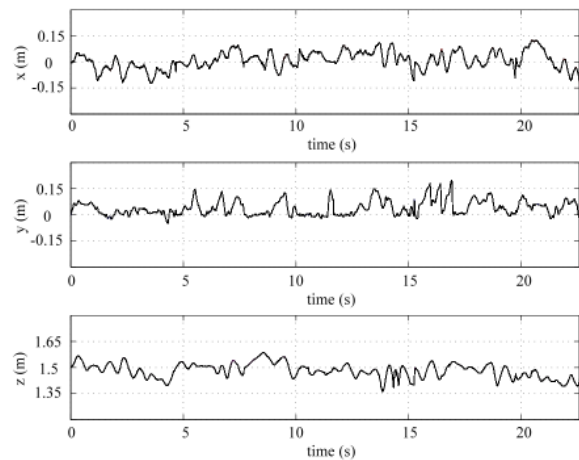

Fig. 17. Hover stabilization experiment: The translational position of the quad-rotorcraft shows that the drift is effectively eliminated, and that the altitude is well stabilized.

Regarding the second experiment (forward flight at constant velocity), the vehicle is first set on manual mode and is positioned exactly over the road model. Then, the automatic velocity regulation is activated from the Ground Station PC. This allows the vision-based algorithm to take control of the quad-rotorcraft. The set of Figures 20, 21, and 22 show the behavior of the helicopter position, velocities, and Euler angles, respectively, for the velocity regulation experiment. It can be seen in Figure 20 that the lateral position and altitude are effectively stabilized. Figure 21 shows how the lateral and altitude velocities are kept near zero while the forward velocity converges to the desired value, which was set to 0.25 m/s. Finally, Figure 22 shows that the roll and yaw angles are kept small, while the pitch angle converges to some positive angle which allows producing the forward displacement at constant velocity. Figure 1 shows an image
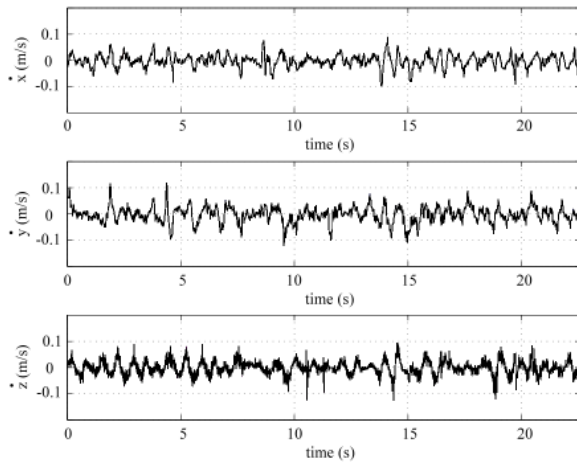
Fig. 18. Hover stabilization experiment: Translational velocities of the quad-rotorcraft.
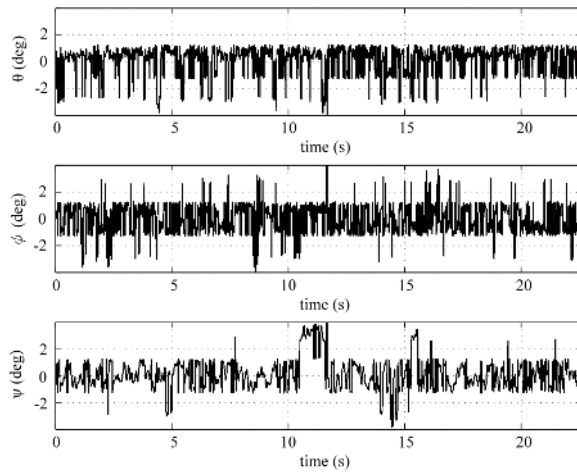

Fig. 19. Hover stabilization experiment: The Euler angles dynamics are not affected by the controller acting on the position stabilization.

of the quad-rotorcraft while performing the forward velocity regulation experiment. A video of this second experiment can be seen at

(http://youtu.be/PpUW9a3S3GQ).

IX. CONCLUSIONS

This paper addressed the problem of stabilizing the 3-dimensional position and translational velocity of a quad-rotorcraft during autonomous flights. The proposed solution, which is based on a vision algorithm for line detection and optical flow computation, uses images furnished by a monocular camera system installed onboard the UAV. Our investigation involves two fundamental characteristics for any autonomous navigation system: we seek to accurately measure translational displacements, while eliminating the position drift when hovering. If the translational drift is correctly compensated, the hover flight can be used as an intermediary task between different flying behaviors, each one of them adapted to different conditions of the environment. Velocity regulation was implemented to establish the different flying
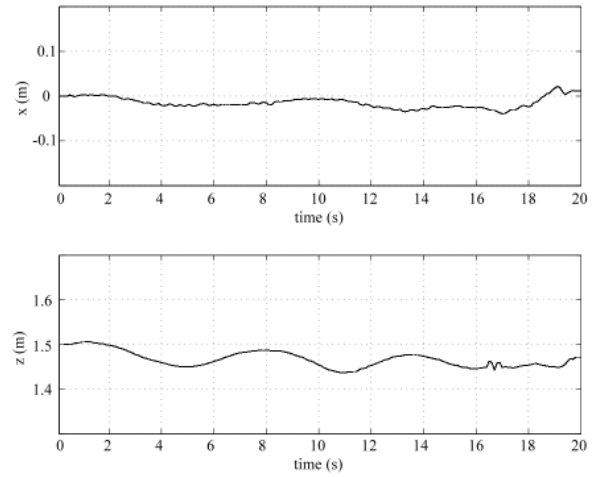

Fig. 20. Forward velocity regulation experiment: The translational position of the quad-rotorcraft. Notice that the lateral position x and the altitude z present only small variations.
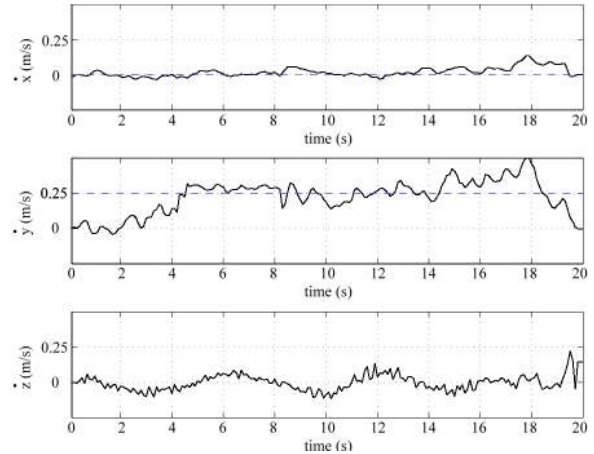

Fig. 21. Forward velocity regulation experiment: The translational velocities during the autonomous flight show how the lateral and altitude velocities are kept near zero while the forward velocity converges to the desired value, which was set to 0.25 m/s.
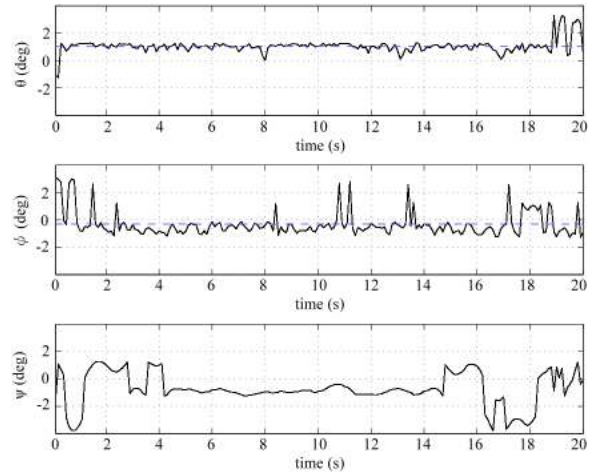

Fig. 22. Forward velocity regulation experiment: The pitch angle converges to some positive angle which allows producing the forward displacement at constant velocity. The roll and yaw angles are kept small.

modes, such as lateral displacement and forward displacement actions. In order to make use of the optical flow in a very appropriate manner, a vision-based altitude controller has also been developed. The combination of these three vision-based controllers (hover flight, velocity regulation, and altitude stabilization) allowed the vehicle to navigate autonomously over a road model in a simulated environment a well as in a real-time application. Two different kinds of missions were effectively tested: position hold over a road segment and road following at constant velocity.

Future work will consist of detecting and avoiding obstacles appearing on a frontal camera when forward flight at constant velocity mode is active. In addition, the velocity regulation algorithm will be tested in combination with a new altitude estimator in order to fly at constant velocity in other environments.

REFERENCES

[1] L.R. Garcia Carríllo, A. Dzul and R. Lozano, *Hovering quad-rotor control: A comparison of nonlinear controllers using visual feedback*, IEEE Transactions on Aerospace and Electronic Systems, vol. 48, issue 4, pp. 3159-3170, October 2012.

[2] S. Salazar, J. Escareno, D. Lara, and R. Lozano, *Embedded control system for a four rotor UAV*, International Journal of Adaptive Control and Signal Processing, vol. 21, no. 2-3, pp. 189-204, April 2007.

[3] A. Johnson, S. Montgomery, and L. Matthies, *Vision guided landing of an autonomous helicopter in hazardous terrain*, International Conference on Robotics and Automation, pp. 4470-4475, Barcelona, Spain, April 2005.

[4] Z. Yu, K. Nonami, J. Shin, and D. Celestino, *3d vision based landing control of a small scale autonomous helicopter*, International Journal of Advanced Robotic Systems, vol. 4, no. 1, pp. 51-56, 2007.

[5] S. Hrabar, G. S. Sukhatme, P. Corke, K. Usher, and J. Roberts, *Combined optic-flow and stereo-based navigation of urban canyons for a UAV*, International Conference on Intelligent Robots and Systems, pp. 302-309, Alberta, Canada, August 2005.

[6] L. Heng, L. Meier, P. Tanskanen, F. Fraundorfer, and M. Pollefeys, *Autonomous Obstacle Avoidance and Maneuvering on a Vision-Guided MAV Using On-Board Processing*, IEEE International Conference on Robotics and Automation, pp. 2472-2477, Shanghai, China, May 9-13, 2011.

[7] E. Rondón, S. Salazar, J. Escareno, and R. Lozano, *Vision-based position control of a two-rotor VTOL miniUAV*, Journal of Intelligent and Robotic Systems, vol. 57, no. 1-4, pp. 49-64, January 2010.

[8] N. Guenard, T. Hamel, and R. Mahony, *A practical visual servo control for an unmanned aerial vehicle*, IEEE Transactions on Robotics, vol. 24, no. 2, pp. 331-341, April 2008.

[9] J. Chahl, M. Srinivasan, and S. Zhang, *Landing strategies in honeybees and applications to uninhabited airborne vehicles*, Int. J. of Robotics Research, vol. 23, no. 2, pp. 101-110, February 2004.

[10] B. Herissé, F. Russotto, T. Hamel, and R. Mahony, *Hovering flight and vertical landing control of a VTOL unmanned aerial vehicle using optical flow*, International Conference on Intelligent Robots and Systems, pp. 801-806, Nice, France, September 2008.

[11] F. Ruffier and N. Franceschini, *Optic flow regulation: the key to aircraft automatic guidance*, Robotics and Autonomous Systems, vol. 50, no. 4, pp. 177-194, March 2005.

[12] M.A. Garrat and J.S. Chahl, *Vision-based terrain following for an unmanned rotorcraft*, Journal of Field Robotics, vol. 25, no. 4-5, pp. 284-301, May 2008.

[13] B. Herissé, T. Hamel, R. Mahony, and F. Russotto, *A nonlinear terrain-following controller for a VTOL unmanned aerial vehicle using translational optical flow*, International Conference on Robotics and Automation, pp. 3251-3257, Kobe, Japan, May 12-17, 2009.

[14] J. Serres, D. Dray, F. Ruffier, and N. Franceschini, *A vision-based autopilot for a miniature air vehicle: Joint speed control and lateral obstacle avoidance*, Autonomous Robots, vol. 25, pp. 103-122, 2008.

[15] J.C. Zufferey, A. Klaptocz, A. Beyeler, J.D. Nicoud, and D. Floreano, *A 10-gram Microflyer for Vision-based Indoor Navigation*, International Conference on Intelligent Robots and Systems, Beijing, China, 2006.

[16] E. Rondón, I. Fantoni-Coichot, A. Sánchez, and G. Sanahuja, *Optical flow-based controller for reactive and relative navigation dedicated to a four rotor rotorcraft*, International Conference on Intelligent Robots and Systems (IROS), pp. 684-689, St. Louis, USA, October 10-15, 2009.

[17] E. Rondón, L.R. Garcia-Carríllo, I. Fantoni, *Vision-Based Altitude, Position and Speed Regulation of a Quadrotor Rotorcraft*, International Conference on Intelligent Robots and Systems (IROS), pp. 628-633, Taipei, Taiwan, October 2010.

[18] J. Shi and C. Tomasi. *Good features to track*, In proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 593-600, Seattle, USA, 1994.

[19] J.Y. Bouguet, *Pyramidal implementation of the Lucas Kanade feature tracker*, Technical report, Intel Corporation, Microprocessor Research Labs, Technical report, 1999.

[20] J.L. Barron, D.J. Fleet and S.S. Beauchemin, *Performance of optical flow techniques*, International Journal of Computer Vision, vol. 12, pp. 43-77, 1994.

[21] S. Umeyama, *Least-Squares Estimation of Transformation Parameters Between Two Point Patterns*, IEEE Transactions on Pattern Analysis and Machine Intelligence, pp. 376-380, April, 1991.

[22] P. Castillo, R. Lozano and A. Dzul, *Modelling and Control of Mini-Flying Machines*, 1st Edition, Springer, June 2005.

[23] T.J. Koo and S. Sastry, *Output tracking control design of a helicopter model based on approximate linearization*, IEEE Conference on Decision and Control, Tampa, FL, December 1998.

[24] G. Sanahuja, *Embeded position estimation and control of a UAV using computer vision*, Ph.D. Thesis, University of Technology of Compiègne, January 2010.

[25] G. Bradski and A. Kaehler, *Learning OpenCV*, 1st Edition, O'Reilly Media Inc. Sebastopol, CA, USA, 2008.

[26] A. Sanchez, L.R. García Carrillo, E. Rondón, R. Lozano and O. García, *Hovering flight improvement of a quad-rotor mini UAV using brushless DC motors*, Journal of Intelligent and Robotic Systems, vol. 61, issue 1-4, pp. 85-101, January 2011.