Robotica (2012) volume 30, pp. 1203–1212. © Cambridge University Press 2012
doi:10.1017/S026357471200001X

Visual servoing applied to real-time stabilization of a multi-rotor UAV

Hugo Romero†,‡,*, Sergio Salazar‡ and Rogelio Lozano‡,§

†*CITIS-UAEH, Area Académica de Computación, 42184 Pachuca Hidalgo, México*
‡*UMI CNRS CINVESTAV, Av. IPN 2508 Col. San Pedro Zacatenco 07360 México D.F., México*
§*UTC, Heudiasyc Centre de Recherches Royallieu BP 20529 60205 Compiègne Cedex France*

(Accepted January 6, 2012. First published online: February 13, 2012)

SUMMARY
In this paper we address the problem of stabilization and local positioning of a four-rotor rotorcraft using computer vision. Our approaches to estimate the orientation and position of the rotorcraft combine the measurements from an Inertial Measurement Unit (IMU) and a vision system composed of a single camera. In the first stage, the vision system is used to estimate the position and yaw angle of the rotorcraft, while in the second stage the vision system is used to estimate the translational velocity of the flying robot. In both cases the IMU gives the pitch and roll angles at a higher rate. The technique used to estimate the position of the rotorcraft in the first stage combines the *homogeneous transformation* approach for the camera calibration process with the *plane-based pose* method for estimating the position. In the second stage, a navigation system using the optical flow is also developed to estimate the translational velocity of the aircraft. We present real-time experiments of stabilization and location of a four-rotor rotorcraft.

KEYWORDS: Visual servoing; Navigation system; UAVs.

1. Introduction

Unmanned Aerial Vehicles (UAVs) are mainly used for assessing situation awareness and surveillance, which are very important issues. These applications require UAVs capable of carrying out hover and navigation, and the performance strongly depends on the applied control strategy and a set of on-board sensors.[1,2] The design of UAVs usually involves the trade-off between performance, price, weight and pay-load which is difficult to achieve.

The standard sensor for measuring orientation in UAVs is Inertial Measurement Unit (IMU) composed of gyros, accelerometers and magnetometers. Outdoor position measurement is usually obtained from Global Positioning System (GPS), while laser radars and ultrasonic sensors are used to avoid obstacles. Nevertheless, the standard techniques based on GPS for navigation are not reliable in urban or hostile environments because the communication signals with the satellites can be easily blocked or jammed, which implies an erroneous location estimation of the flying machine. The positioning systems based on radar or laser add a considerable weight on the UAV and they emit external signals easily detectable in hostile environments.

An alternative way for measuring position is based on computer vision. Vision systems have many advantages with respect to typical sensors: they are passive, i.e. do not emit any external signals, and they can be small and light. In addition, a single camera is a low-cost device that provides a large amount of information about the surrounding non-structured environment.

Vision systems can be used in UAVs for navigating indoor and in urban environments. The improving quality of digital imaging and the increasing performance of image treatment are such that computer vision is an affordable approach to estimate location and speed of a flying machine.

Earlier works have shown the great potential for vision systems applied to estimate UAV position.[3–7] In ref. [3] Wu *et al.* present a vision-aided inertial navigation approach based on tracking a rectangular target (dark window against white background). Camera is used to measure the centroid of the target and its projected area in the image. This visual information, together with heading measurements from a magnetometer, is processed by an extended Kalman filter (EKF) to update the state of the vehicle and correct errors produced when integrating IMU signals. This approach is sensitive to variations in magnetometer measurements, and its accuracy reduces with distance from the target.

Mejias *et al.*[4] have developed a vision system based on two-dimensional (2D) visual servoing to locate UAV with respect to urban features (windows), which are chosen by the user. Since this work is based on 2D visual servoing, the position of helicopter is estimated using image coordinates, which means that the position of the helicopter is determined by position in the image plane of the selected urban feature. Then, in order to arrive a desired position, the image of the selected urban feature should be positioned in specific image coordinates.

The autonomous helicopter reported by Yoshihata *et al.*[5] shows a way to localize the helicopter when the target is partially occluded. They used an auto stable coaxial helicopter and a couple of outboard cameras. A strong assumption made in this work is that the four coordinate systems used should be parallel at least in one axis. In a work closely related to ours,[6] Saripalli *et al.* present the autonomous landing of a UAV based on vision; they assume that the target used (helipad) is plane and helicopter is

* Corresponding author. E-mail: hromero72@gmail.com

https://doi.org/10.1017/S026357471200001X Published online by Cambridge University Press

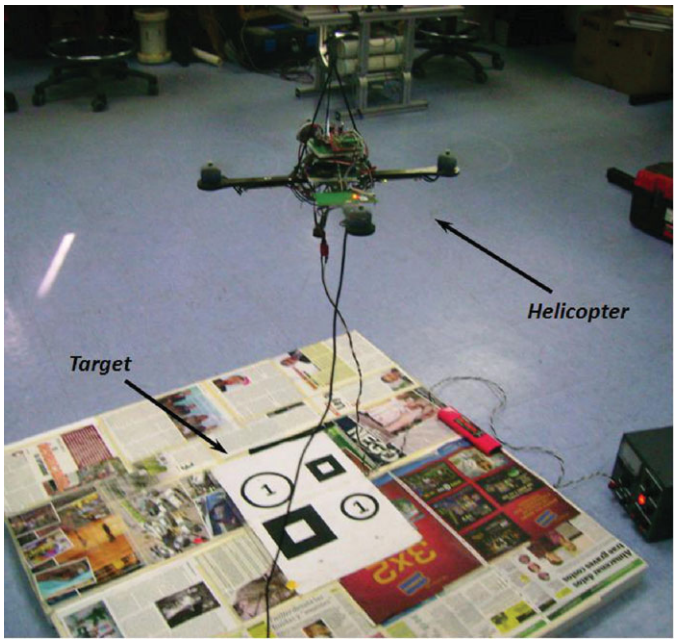


Fig. 1. (Colour online) Autonomous flight of X4 using computer vision.



Fig. 2. (Colour online) (a) Four rotors platform; (b) target used composed of two pairs of different symbols; (c) segmented image; (d) graphical representation of optical flow vector corresponding to camera movement from left to right.

equipped with an ultrasonic sonar like a complementary sensor.

Other previous works using four rotor helicopters as a platform are as given in refs. [8–10]. Altug *et al.*,[8] developed a visual servoing control for a four-rotor airfcraft using a couple of cameras, one of these cameras on board and the other located on the ground. A strong assumption in this work is that at every moment of the flight, the cameras observe each other. Bourquardez *et al.*[9] used an image-based visual servoing technique in order to achieve the autonomous flight of a Quad rotor mini-helicopter. This kind of visual servoing technique is more robust with respect to uncertainties in image segmentation and avoids the problem of error in camera calibration procedure. However, it only works well in a nearby region of a desired position. Herisse *et al.*[10] used only the optical flow and an IMU module to develop a hovering stable flight and automatic landing.

The approach presented in the present paper combines a pose estimation method and an optical flow approach with an IMU module to estimate the whole state of our flying robot. Our approach uses a 3D visual servoing, which implies that real-world measures are used to localize the UAV. With a simple pose estimation algorithm we are able to estimate the real 3D position $(x,\ y,\ z)$ and the heading (ψ) of a mini-helicopter without additional sensor, while an optical flow approach is applied to estimate the translational velocity $\dot{x},\ \dot{y}$. Furthermore, the vision platform is built with a single light onboard wireless camera. Moreover, the control law applied in our case is more adapted to control this kind of dynamical system because it privileges the attitude control (θ and ϕ angles) over position control. The proposed approach is an alternative approach to achieve autonomous hover flight of four-rotor rotorcraft (see Fig. 1).

Combination of inertial and vision sensors increases the capabilities of robotic systems and enlarges the potential application of vision systems. The advantage of combining these two sensor modalities is the complementary characteristics of cameras and inertial sensors.[11,12] On one hand, the inertial sensors have large measurement uncertainty at slow motion and lower relative uncertainty at high velocities. Inertial sensors can measure very high velocities and accelerations. On the other hand, cameras can track features very accurately at low velocities.

We have built a multi-rate experimental platform based on a four-rotor rotorcraft, including a Rabbit microprocessor, an IMU and a single camera. This platform runs at two different sampling rates: the control law and the IMU run at 60 Hz, while the vision system runs at 15 Hz. The vision system provides an estimation of the position, the translational speed and the heading of the rotorcraft. It is clear that the performance of a control law depends heavily on the precision of the sensors that are used to estimate the orientation and position of the autonomous vehicle.

In this paper we have applied a nonlinear control strategy for stabilizing a four-rotor aircraft by using the attitude, position and velocity estimates by an appropriate fusion of the information provided by inertial sensors and the information obtained from the vision system. First, we present a technique to localize the aircraft with respect to a target composed of two pairs of different symbols centered in the vertices of a square on the floor of the laboratory (see Fig. 2). The technique combines *the homogeneous transformation approach* as camera calibration method and the *plane-based pose method* to estimate the position of the aircraft. The pitch and roll angular measurements are obtained from the IMU, while the x–y position and the yaw angle are provided by the vision system. We illustrate the performance of the proposed vision-based control scheme in a real-time experiment.

The second part of the paper is focused on the use of vision to estimate the horizontal velocity of a four-rotor rotorcraft with respect to the ground using optical flow. The objective is

https://doi.org/10.1017/S026357471200001X Published online by Cambridge University Press

to stabilize the rotorcraft at hover around a desired set-point. The optical flow field or image velocity field is computed over a sequence of time-ordered images that allow the estimation of projected 2D image motion as either instantaneous image velocity or discrete image displacements.[13]

This paper is organized as follows: In Section 2, notation used through the paper is given. Section 3 shows the pose estimation method applied to estimate the 3D position and orientation of the camera. The strategy to estimate the translational velocity of a rotorcraft using optical flow is described in Section 4. The mathematical nonlinear model using the Euler–Lagrange approach of a multi-rotor mini-rotorcraft is presented in Section 5. The control strategy to stabilize at hover the multi-rotor platform is introduced in Section 6. Section 7 shows and describes the platform architecture used in this experiment. Finally, experimental results and conclusions are presented in Sections 8 and 9, respectively.

2. Notation

In order to facilitate the reading of this work, following notations are introduced:

$\mathbb{T}$	Projection matrix
$\mathbf{K}$	Intrinsic parameters matrix
$\mathbf{R}, \mathbf{S}, \mathbf{N}$	Rotation matrices
$\bar{\mathbf{R}}, \bar{\mathbf{S}}, \bar{\mathbf{N}}$	3×2 rotation submatrices
$\mathbf{t}, \mathbf{v}, \mathbf{w}$	Translation vectors
$p = [x, y, z]^T$	3D real point
$q = [\rho, \gamma]^T$	Image point
v	Translational velocity
ω	Angular velocity
$\mathbf{OF}$	Optical flow
μ	Optimal scale factor
τ	Generalized momentum
$\mathbf{F}_e$	Translational force applied to helicopter
$\mathbf{J}$	Inertia matrix
$\mathbf{C}$	Coriolis forces vector
$\mathbf{F}_b$	Thrust vector of the helicopter
g	Gravitational acceleration
m	Mass of the helicopter
$\mathbf{u}$	Control input
θ, ϕ, ψ	Pitch, roll and yaw angles

3. Position and Orientation Estimation Approach

In this section we present the method implemented to estimate the camera orientation and location. Pose estimation method used in this paper is a 3D visual servoing strategy because the pose estimation is computed in a 3D Cartesian space. The pose of the camera with respect to the target (and *vice versa*) is estimated from image features corresponding to the perspective projection of target in the image. Therefore, it is based on the knowledge of a perfect geometry model of the target and calibrated camera. The 3D visual servoing algorithms face up the problem related to errors in the calibration data and the image segmentation. However, our control strategy compensates those errors. The 2D visual servoing techniques have no such problems because the pose estimation is computed in the 2D image space. However, its convergence is theoretically ensured only in a region around the desired position, and it is not global as in the case of 3D visual servoing.

Camera calibration procedure is a necessary stage before applying the pose estimation method used in this work. Camera model considered is the perspective projection, where the projection of 3D homogeneous points $p = (x,\ y,\ z,\ 1)^T$ in their respective image homogeneous point $q = (\rho,\ \gamma,\ 1)^T$ is defined by $q = \mathbb{T}p$ with the 3×4 projection matrix $\mathbb{T}$ given as

$$\mathbb{T} = \mathbf{KR}\,[\mathbf{I} \mid -\mathbf{t}]\,, \tag{1}$$

where $\mathbf{K}$ is a usual 3×3 calibration matrix with intrinsic parameters of camera, $\mathbf{R}$ is a 3×3 orthogonal matrix representing camera orientation and $\mathbf{t}$ is a 3×1 vector containing the camera position.[14,15]

After applying the calibration procedure to obtain the intrinsic and extrinsic parameters of the camera, we can proceed to estimate the position and orientation of the camera with respect to a target composed of two pairs of different symbols. Each one of these pairs of symbols has a different size and are posed in the vertices of a square whose geometry is well known. In order to compute the gravity center of each symbol in the scene, an object detection algorithm is applied, which provides robustness with respect to possible changes in the luminosity of the scene.[16,17] The position and orientation estimation algorithm related to the calibration method is described in the following subsection.

3.1. Plane-based pose method

We now describe the coordinate transformations that lead from 3D points on a plane $\mathbb{P}$ in the real world to the coordinates of their projections in the image plane.[18] Let $p_i^{\mathcal{P}}$ be the ith real point on plane $\mathbb{P}$ having homogeneous coordinates $(x_i, y_i, z_i, 1)^T$ with respect to a local plane frame $\mathcal{P}$. Let the position and orientation of plane $\mathbb{P}$ be given by a rotation matrix $\mathbf{S}$ and translation vector $\mathbf{v}$ with respect to a global 3D world reference frame $\mathcal{I}$. In this global reference frame, p_i is expressed as

$$p_i^{\mathcal{I}} = \begin{bmatrix} \mathbf{S}_{3\times3} & \mathbf{v}_{3\times1} \\ \mathbf{0}_{1\times3} & 1 \end{bmatrix} p_i^{\mathcal{P}}, \tag{2}$$

with $z_i = 0$. Let the position of the camera be given by the rotation matrix $\mathbf{N}$ and the translation vector $\mathbf{w}$ with respect to global frame $\mathcal{I}$. Therefore, the coordinates of p_i in the local camera frame $\mathcal{C}$ are

$$p_i^{\mathcal{C}} = \begin{bmatrix} \mathbf{N}_{3\times3} & \mathbf{w}_{3\times1} \\ \mathbf{0}_{1\times3} & 1 \end{bmatrix} p_i^{\mathcal{I}}. \tag{3}$$

These two coordinate transformations are graphically represented in Fig. 3. Combining Eqs. (1), (2) and (3), the coordinates of the corresponding projected point in the image

https://doi.org/10.1017/S026357471200001X Published online by Cambridge University Press

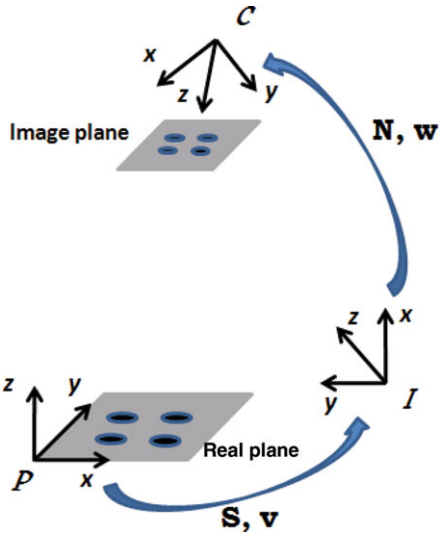


Fig. 3. (Colour online) Graphical representation of coordinate transformations between general coordinate frame $\mathcal{I}$, the camera frame $\mathcal{C}$ and the plane frame $\mathcal{P}$.

plane coordinates are

$$\begin{aligned} q_i &= \begin{bmatrix}\mathbf{K}\ \mathbf{0}\end{bmatrix} p_i^{\mathcal{C}} \\ &= \begin{bmatrix}\mathbf{K}\ \mathbf{0}\end{bmatrix} \begin{bmatrix}\mathbf{N}\ \mathbf{w} \\ \mathbf{0}\ 1\end{bmatrix} \begin{bmatrix}\mathbf{S}\ \mathbf{v} \\ \mathbf{0}\ 1\end{bmatrix} p_i^{\mathcal{P}} \\ &= \mathbf{K}\begin{bmatrix}\mathbf{NS}\ \mathbf{Nv}+\mathbf{w}\end{bmatrix} (x_i\ \ y_i\ \ 0\ \ 1)^T . \end{aligned}$$

The transformation $\mathbb{T}$ is defined as

$$\mathbb{T} = \mathbf{K}\left[\left(\mathbf{N}\bar{\mathbf{S}}\right)_{3\times 2}\ (\mathbf{Nv}+\mathbf{w})_{3\times 1}\right],$$

where $\bar{\mathbf{S}}$ is the 3×2 submatrix of $\mathbf{S}$ consisting of its first two columns. Since the calibration is known, we can compute

$$\mathbf{A} = \mathbf{K}^{-1}\mathbb{T} = \begin{bmatrix}\mathbf{N}\bar{\mathbf{S}}\ \mathbf{Nv}+\mathbf{w}\end{bmatrix}. \tag{4}$$

Defining the total rotational matrix $\mathbf{R}$ and the total translational vector $\mathbf{t}$ as

$$\begin{aligned} \mathbf{R} &= \mathbf{NS}, \\ \mathbf{t} &= \mathbf{S}^T\mathbf{v} + \mathbf{S}^T\mathbf{N}^T\mathbf{w}, \end{aligned}$$

Eq. (4) can be rewritten as

$$\mathbf{A} = \mathbf{RA}_1 = \mathbf{R}\begin{bmatrix}\mathbf{I}_{2\times 2}\mathbf{t} \\ \mathbf{0}\end{bmatrix}.$$

In the general case, $\mathbf{A}$ is obtained from the following minimization problem:

$$\min_{\mathbf{R},\mathbf{t},\mu} \|\mu\mathbf{A} - \mathbf{RA}_1\|_F^2, \quad \text{subject to } \mathbf{R}^T\mathbf{R} = \mathbf{I}_{3\times 3},$$

where $\|\cdot\|_F$ denotes the Frobenius norm. As shown in ref. [18], the solution for $\mathbf{R}$ does not depend on μ and $\mathbf{t}$, and thus the optimal solution for $\mathbf{R}$ can be defined as

$$\min_{\bar{N}} \left\|\bar{\mathbf{A}} - \bar{\mathbf{R}}\right\|_F^2 \text{ , subject to } \bar{\mathbf{R}}^T\bar{\mathbf{R}} = \mathbf{I}_{2\times 2}. \tag{5}$$

In order to find the optimal solution of Eq. (5), we compute the Singular value decomposition (SVD) of $\bar{\mathbf{A}} = \mathbf{U}_{3\times 2}\mathbf{D}_{2\times 2}\mathbf{V}_{2\times 2}$, then $\bar{\mathbf{R}}$ is given by

$$\bar{\mathbf{N}} = \mathbf{UV}.$$

The third column of the rotation matrix $\mathbf{R}$ can be computed from the cross product of the columns of $\bar{\mathbf{R}}$. Once $\mathbf{R}$ is computed, the optimal scale factor μ and vector $\mathbf{t}$ are determined by

$$\mu = \frac{tr(\bar{\mathbf{R}}^T\bar{\mathbf{A}}^T)}{tr(\bar{\mathbf{A}}^T\bar{\mathbf{A}}^T)},$$

$$\mathbf{t} = \mathbf{R}^T\mathbf{A}\begin{pmatrix}0\\0\\\mu\end{pmatrix}.$$

With these relations we obtain the position ($\mathbf{t}$ vector) and orientation (from $\mathbf{R}$ matrix) of the camera with respect to the plane located in a well-known coordinate system $\mathcal{I}$.

In order to compute the vertical velocity $\dot{z}$, which is required for the implementation of control law described in Section 6, we have implemented an approximation of the derivative using two consecutive altitude measurements provided by the vision system. This approximation can be expressed as follows:

$$\dot{z}(t) = \frac{z(t) - z(t-\Delta t_v)}{\Delta t_v}, \tag{6}$$

where Δt_v is the sampling period of the vision system. Once the vertical velocity is estimated, a second-order low pass filter is applied to remove the inherent noise of this approximation.

4. Optical Flow-Based Control

There exist numerous computational models for estimating image velocity, which are described in ref. [19]. Optical flow measurement methods can be classified as intensity-based differential methods, frequency-based filtering methods or correlation-based methods. In addition, there exist methods for the computation of discontinuous or multiple-valued optical flow and techniques for performing temporal refinements of motion estimate as more information becomes available through the image acquisition process. In this paper we have used the pyramidal Lucas–Kanade method proposed in ref. [20], which is an intensity-based differential method.

Let I_{t_1} and I_{t_2} be two gray scale images taken at two different time instants and represented as discrete functions that belong to $\mathbb{R}^{m_\rho * n_\gamma}$. In addition, the gray scale values for a particular pixel $q = (\rho,\ \gamma)^T$ in these two images are defined as $I_{t_1}(q) = I_{t_1}(\rho_q, \gamma_q)$ and $I_{t_2}(q) = I_{t_2}(\rho_q, \gamma_q)$, where ρ_q and γ_q are the pixel coordinates (row and column, respectively)

https://doi.org/10.1017/S026357471200001X Published online by Cambridge University Press

of a generic image point q. Considering an image point $q_{t_1} = [\rho_{q_1},\ \gamma_{q_1}]$ in the first image I_{t_1}, the objective is to find the same image point in the second image I_{t_2} defined as $q_{t_2} = [\rho_{q_2},\ \gamma_{q_2}]$ such that

$$q_{t_2} = q_{t_1} + r = [\rho_{q_1} + r_\rho,\ \gamma_{q_1} + r_\gamma]$$

such that $I_{t_1}(q_1)$ is equal to $I_{t_2}(q_2)$. The vector $r = [r_\rho,\ r_\gamma]$ defines the image velocity or optical flow and it minimizes the following residual function

$$\epsilon(r) = \sum_{\rho_q = rho_{q_1} - u_\rho}^{\rho_{q_1}+u_\rho} \sum_{\gamma_q = \gamma_{q_2} - u_\gamma}^{\gamma_{q_2}+u_\gamma} (D(\rho_q, \gamma_q, r))^2$$

with

$$D(\cdot) = I_{t_1}(\rho_q, \gamma_q) - I_{t_2}(\rho_q + r_\rho,\ \gamma_q + r_\gamma),$$

where u_ρ and u_γ are two integers that define the integration window size, i.e. $(2u_\rho + 1) \times (2u_\gamma + 1)$. There is a natural trade-off between local accuracy and robustness when choosing the size of the integration window, which is known as the *aperture problem*. On the one hand, a large integration window is required to estimate high speeds but it increases the chance to obtain points erroneously correlated. On the other hand, a small integration window reduces optical flow computing time and minimizes the chances of having incorrect correlation of image points in two consecutive images. Typical values for u_ρ and u_γ lie in the interval (2, 7). In this paper we consider an iterative implementation of the Lucas–Kanade optical flow algorithm described in ref. [20], which is easy to implement and it has an adaptive integration window. This method is based on a pyramid representation of a generic image. The central motivation behind the pyramidal representation is to be able to handle large pixel motions. This representation acts as a low pass filter. The maximum number of pyramidal images that we have used is $\mathbb{L} = 3$. See ref. [20] for a complete description of this method.

Optical flow is the apparent motion of features in the field of view of a moving observer. This flow can be created by two kinds of observer motion: translational ($\mathbf{F}_t$) and rotational ($\mathbf{F}_r$). The optical flow (**OF**) experienced by a camera moving with translational velocity v and angular velocity ω while viewing an object at distance **d** and offset β from the direction of travel is defined in ref. [21] and can be expressed as follows:

$$\mathbf{OF} = \frac{v}{\mathbf{d}} \sin\beta - \omega. \tag{7}$$

The first term on the right-hand side represents the optical flow produced by the translational motion. Large amounts of optical flow are produced when **d** is small and the maximal flow is generated when the plane that contains the features is perpendicular to translational motion direction ($\beta = 90^\circ$). An optical flow basic scheme is shown in Figure 4, where all variables are represented graphically. According to Eq. (7), the velocities experienced by the aerial robot in x and y directions can be expressed, respectively, as follows:

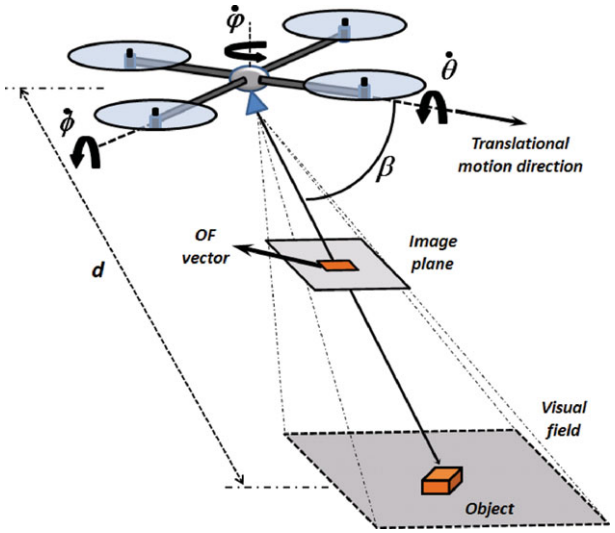


Fig. 4. (Colour online) Basic scheme of optical flow.

$$\dot{x} = \frac{(\mathbf{OF}_x + \dot{\theta})}{\sin\beta}\, z, \tag{8}$$

$$\dot{y} = \frac{(\mathbf{OF}_y + \dot{\phi})}{\sin\beta}\, z. \tag{9}$$

Notice that singularities in the above equations appear when $\beta = 0^\circ$. Nevertheless, in practice the roll and pitch angles lie at an interval of $\pm 5^\circ$, therefore we have $85^\circ < \beta < 95^\circ$ when the mini-helicopter is appropriately stabilized. The estimation of the angular speeds $\dot{\theta}$ and $\dot{\phi}$ are obtained by using the gyro information for each axis. Furthermore, this information is used in the vision system to compensate the rotational speed component of the optical flow estimation. The translational velocities $\dot{x}$ and $\dot{y}$ are updated at the sampling time of the vision system using the angular rate information. A second-order low-pass filter has been also applied to cancel noise generated by the mechanical vibrations of a mini four-rotor rotorcraft that affect the estimation of β and generate false measurements of optical flow.

5. Four-rotor Rotorcraft Model

In this work we use a four-rotor mini-helicopter whose parameters are $m = 0.5\,\text{kg}$, $l = 0.20\,\text{m}$. This type of configuration has many advantages over conventional helicopters and other configurations. The chosen configuration does not have a swashplate and it does not need any blade pitch control, which simplifies the design of the vehicle. The gyroscopic effects and aerodynamic torques tend to cancel in trimmed flight because the front and rear motors rotate counterclockwise while the right and left motors rotate clockwise. Furthermore, the use of four rotors allows each individual rotor to have a smaller diameter than the equivalent helicopter rotor for a given vehicle size, producing less kinetic energy in the rotors during flight. This property

https://doi.org/10.1017/S026357471200001X Published online by Cambridge University Press

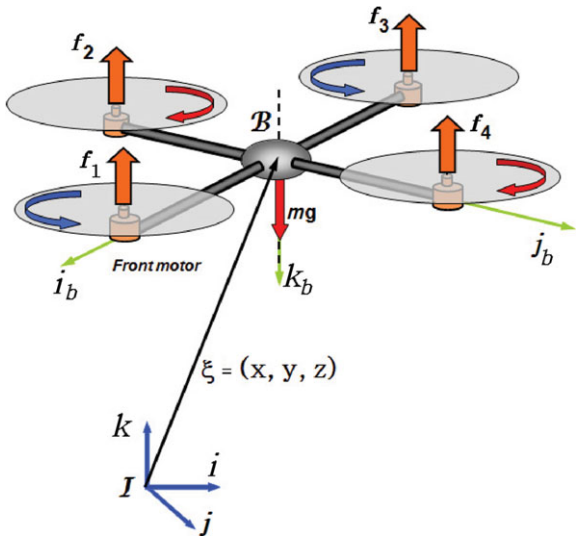


Fig. 5. (Colour online) Four-rotor rotorcraft scheme.

reduces the possible damage to persons caused by the rotors.

A four-rotor rotorcraft (Fig. 5) is controlled by varying the angular speed of each of the rotors. The force f_i produced by motor i is proportional to the square of the angular speed, that is $f_i = k\omega_i^2$. The main thrust is the sum of the thrusts of each motor. The pitch torque is a function of the difference $f_1 - f_3$, while the roll torque is a function of $f_2 - f_4$, and the yaw torque is the sum $\tau_{M_1} + \tau_{M_2} + \tau_{M_3} + \tau_{M_4}$, where τ_{M_i} is the reaction torque of motor i due to shaft acceleration and the blade's drag. Terms $\mathcal{I} = \{\hat{i}, \hat{j}, \hat{k}\}$ and $\mathcal{B} = \{\hat{i}_b, \hat{j}_b, \hat{k}_b\}$ denote the inertial and the fixed-body frame, respectively, and $q_e = (x,\ y,\ z,\ \psi,\ \theta,\ \phi)^T = (\xi,\ \eta)^T$ are generalized coordinates that describe the vehicle's position and orientation. $\xi \in \mathbb{R}^3$ denotes the position of the vehicle's center of mass relative to the inertial frame $\mathcal{I}$, and $\eta \in \mathbb{R}^3$ are three Euler angles.

The dynamical model is obtained by using the Euler–Lagrange approach. For simplicity, the equations are separated into translational and rotational displacement as follows:

$$\begin{aligned} \mathbf{F}_e &= m\ddot{\xi} + m\mathbf{g}, \\ \tau &= \mathbf{J}\ddot{\eta} + \dot{\mathbf{J}}\dot{\eta} - \frac{1}{2}\frac{\partial}{\partial \eta}(\dot{\eta}^T \mathbf{J}\,\dot{\eta}), \end{aligned} \tag{10}$$

where m denotes the mass of the vehicle, $\tau \in \mathbb{R}^3$ denotes the generalized momentum, $\mathbf{F}_e \in \mathbb{R}^3$ is the translational force applied to the rotorcraft, $\mathbf{g} \in \mathbb{R}^3$ denotes acceleration due to the gravity ($\mathbf{g} = [0\hat{i},\ 0\hat{j},\ -g\hat{k}]^T$) and $\mathbf{J} \in \mathbb{R}^{3x3}$ represents the inertia matrix, which is symmetric and positive definite. The Coriolis, gyroscopic and centrifugal terms are expressed as

$$\mathbf{C}(\eta, \dot{\eta})\,\dot{\eta} \triangleq \dot{\mathbf{J}}\dot{\eta} - \tfrac{1}{2}\tfrac{\partial}{\partial \eta}(\dot{\eta}^T \mathbf{J}\dot{\eta}). \tag{11}$$

The model Eq. (10) can be rewritten as

$$\begin{aligned} m\ddot{\xi} &= \mathbf{F}_e - m\mathbf{g}, \\ \mathbf{J}\ddot{\eta} + \mathbf{C}(\eta, \dot{\eta})\,\dot{\eta} &= \tau. \end{aligned} \tag{12}$$

The total thrust in the body frame $\mathcal{B}$ is

$$\mathbf{F}_b = \sum_{i=1}^{4} f_i.\hat{k}_b = \sum_{i=1}^{4} k\tilde{\omega}_i^2 \hat{k}_b, \tag{13}$$

where $\tilde{\omega}_i$ is the angular speed of motor i and $k > 0$ is a constant that depends from air density, shape, size of blades etc. Let us define $\mathbf{u} = \sum_{i=1}^{4} k\tilde{\omega}_i^2$ as a collective input, $\mathbf{F}_b$ can also be expressed in the inertial frame $\mathcal{I}$ as $\mathbf{F}_e = \mathbf{R}_e^b \mathbf{F}_b$, where $\mathbf{R}_e^b$ is a rotation matrix described in ref. [22]. The total momentum is given by

$$\tau = \tau_\phi \hat{i}_b + \tau_\theta \hat{j}_b + \tau_\psi \hat{k}_b \tag{14}$$

with

$$\left.\begin{aligned} \tau_\phi &= (f_2 - f_4)\,l \\ \tau_\theta &= (f_1 - f_3)l \\ \tau_\psi &= \tau_{M_1} + \tau_{M_2} + \tau_{M_3} + \tau_{M_4} \end{aligned}\right\}, \tag{15}$$

where l is the distance from each motor to the gravity center and τ_{M_i} is the reaction torque of motor i due to shaft acceleration and the blade's drag.

6. Control Strategy

This section presents a control strategy for stabilizing a four-rotor UAV at hover. It has been show in ref. [22] that the closed loop system is stable. Rewriting the control algorithm, the input $\mathbf{u}$ is essentially used to reach a desired altitude value. The control input τ_ψ is then used to set yaw displacement to a desired value ψ_d. τ_θ is used to control the pitch and the horizontal movement in x-axis. Similarly, τ_ϕ is used to control the roll and the horizontal displacement in y-axis. In order to simplify, let us propose the following change of input variables:

$$\tau = \mathbf{J}\widetilde{\tau} + \mathbf{C}(\eta, \dot{\eta})\,\dot{\eta}, \tag{16}$$

where

$$\widetilde{\tau} = \left[\widetilde{\tau}_\phi\ \widetilde{\tau}_\theta\ \widetilde{\tau}_\psi\right]^T. \tag{17}$$

So rewriting Eq. (10), it follows that

$$m\ddot{x} = -\mathbf{u}\sin\theta, \tag{18}$$

$$m\ddot{y} = \mathbf{u}\cos\theta\sin\phi, \tag{19}$$

$$m\ddot{z} = \mathbf{u}\cos\theta\cos\phi - mg, \tag{20}$$

$$\ddot{\phi} = \widetilde{\tau}_\phi, \tag{21}$$

$$\ddot{\theta} = \widetilde{\tau}_\theta, \tag{22}$$

$$\ddot{\psi} = \widetilde{\tau}_\psi, \tag{23}$$

where x and y are the displacements in the horizontal plane, and z is the vertical position. ψ is the yaw angle around z-axis, θ is the pitch angle around y-axis and ϕ is the roll angle around x-axis. The control inputs $\mathbf{u}$, $\widetilde{\tau}_\phi$, $\widetilde{\tau}_\theta$ and $\widetilde{\tau}_\psi$ are the total thrust or collective input, the roll, pitch and yaw angular moments, respectively.

https://doi.org/10.1017/S026357471200001X Published online by Cambridge University Press

6.1. Altitude and yaw control

The control of the vertical position can be obtained by using the following control input:

$$\mathbf{u} = \frac{(r_1 + mg)}{\cos\theta\cos\phi} \tag{24}$$

with

$$r_1 = -a_{z_1}\dot{z} - a_{z_2}(z - z_d), \tag{25}$$

where a_{z_1} and a_{z_2} are positive constants and z_d is the desired altitude. The yaw angular position can be controlled by applying

$$\tilde{\tau}_\psi = -a_{\psi_1}\dot{\psi} - a_{\psi_2}(\psi - \psi_d), \tag{26}$$

where ψ_d is the desired yaw angular position. The control parameters a_{ψ_1}, a_{ψ_2} a_{z1} and a_{z_2} are chosen to ensure a stable, fast and well-damped response in the vertical and yaw axis.

6.2. Lateral position and roll control

Introducing Eqs. (24) and (25) into Eqs. (19) and (21) and using the fact that for a time arbitrarily large but finite, there exists a small upper bound for $|\phi|$ such that the difference $\tan\phi - \phi$ is arbitrarily small. It leads us to obtain

$$\ddot{y} = g\phi, \tag{27}$$

$$\ddot{\phi} = \tilde{\tau}_\phi \tag{28}$$

or

$$y^{(iv)} = g\tilde{\tau}_\phi, \tag{29}$$

which represents four integrators in cascade. Now using the technique developed in ref. [23] based on nested saturations to stabilize four integrators in cascade, the controller is given as follows:

$$\begin{aligned}\tilde{\tau}_\phi = -\sigma_{\phi_1}\big(\dot{\phi} + \sigma_{\phi_2}\big(\dot{\phi} + \phi + \sigma_{\phi_3}\big(\dot{\phi} + 2\phi + \tfrac{\dot{y}}{g}\\ + \sigma_{\phi_4}\big(\dot{\phi} + 3\phi + 3\tfrac{\dot{y}}{g} + \tfrac{y}{g}\big)\big)\big)\big),\end{aligned} \tag{30}$$

where σ_α represents a saturation function as

$$\sigma_\alpha(h) = \begin{cases} \alpha & \text{if } h > \alpha \\ h & \text{if } -\alpha \le h \le \alpha. \\ -\alpha & \text{if } h < -\alpha \end{cases} \tag{31}$$

This control law guaranties convergence to zero of ϕ, $\dot{\phi}$, y and $\dot{y}$.

6.3. Forward position and pitch control

The procedure used in the previous section for roll control is now used for pitch control. The x-θ subsystem can be written as

$$\begin{aligned}\ddot{x} &= -g\tan\theta, \\ \ddot{\theta} &= \tilde{\tau}_\theta.\end{aligned} \tag{32}$$

Proceeding as above, after a finite time, θ is small enough such that $\tan\theta \approx \theta$. Therefore, the subsystem (32) reduces to

$$\begin{aligned}\ddot{x} &= -g\theta, \\ \ddot{\theta} &= \tilde{\tau}_\theta.\end{aligned} \tag{33}$$

Similarly, the roll control input is given by

$$\begin{aligned}\tilde{\tau}_\theta = -\sigma_{\theta_1}\big(\dot{\theta} + \sigma_{\theta_2}\big(\dot{\theta} + \theta + \sigma_{\theta_3}\big(\dot{\theta} + 2\theta - \tfrac{\dot{x}}{g}\\ + \sigma_{\theta_4}\big(\dot{\theta} + 3\theta - 3\tfrac{\dot{x}}{g} - \tfrac{x}{g}\big)\big)\big)\big),\end{aligned} \tag{34}$$

this control law guarantees convergence to zero of θ, $\dot{\theta}$, x and $\dot{x}$.

The optical flow method computes the linear velocities $\dot{y}$ and $\dot{x}$ in Eqs. (30) and (34), respectively, and hence via numeric integration we are able to estimate the position x and y.

7. Platform Architecture

The experimental platform is composed of a four-rotor aircraft with a microprocessor RABBIT RCM3400 on board. This microprocessor has the following main features: module running at 29.4 MHz, 512-K flash memory, 4 pulse width modulation (PWM) outputs, 8 A/D channels, 6 serial ports and 2 input capture channels.

The microprocessor runs the control algorithm in real time to stabilize the flying machine, therefore it reads the information provided by IMU sensors[24] and sends control signals to four motors. The modem is used to receive information about the location, translational speed and yaw angle obtained from the image treatment. The control law described in Section 6 is computed by the microprocessor at every 16.7 ms (60 Hz), which is same for the IMU used to compute (θ, ϕ, $\dot{\theta}$, $\dot{\phi}$ and $\dot{\psi}$). However, the sampling period to estimate the rest of the state (x, y, z, $\dot{x}$, $\dot{y}$, $\dot{z}$ and ψ) is 66.7 ms (15 Hz). It means that the control law uses four times the same values of position, translational velocity and yaw angle until they are updated at the next sampling time of the vision system.

The IMU module is a low-cost inertial measurement unit based on inertial sensors. These inertial signals are acquired using the analog inputs of microprocessor RABBIT 3400. The on-board inertial sensors are three surface micro-machined angular rate sensors (gyroscope ADXRS150) arranged orthogonally and a dual-axis micro-machined silicon accelerometer (ADXL203). The three angular rates ($\dot{\phi}$, $\dot{\theta}$, $\dot{\psi}$) are given by the gyros, and the two angular positions (ϕ, θ) are estimated using a complementary filter.[25]

Data fusion is used to estimate the angular position of the aircraft by appropriately combining accelerometers and gyros measurement signals. The gyro signal θ_g is passed through a filter $F_1(s)$, while the accelerometer measurement θ_a is passed through a filter $F_2(s)$. The estimate $\hat{\theta}$ is given as $\hat{\theta} = F_1(s)\theta_g + F_2(s)\theta_a$. The complementary filters are such that $sF_1(s) + F_2(s) = 1$, where $F_2(s)$ is a first-order low-pass filter $\frac{k}{k+s}$. For $k = 0$, $F_2(s) = 0$, and $F_1(s)$ becomes a

https://doi.org/10.1017/S026357471200001X Published online by Cambridge University Press

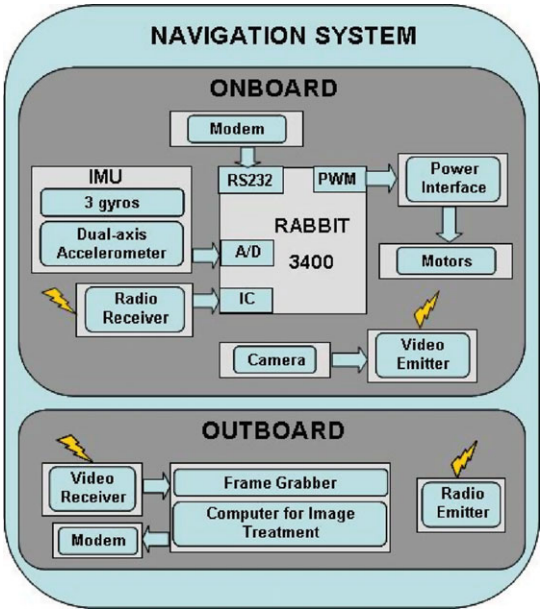


Fig. 6. (Colour online) Architecture scheme of navigation system.

pure integration and $\hat{\theta}$ diverges due to gyro drift. On the other hand, for $k = \infty$, $F_2(s) = 1$, and $F_1(s) = 0$ and the data from the gyro is completely suppressed. In practice, the filter parameter k has been selected to obtain an estimate $\hat{\theta}$, which is as close as possible to the actual value of θ.

The platform for the image treatment consists of a single camera combined with the IMU. The image is captured by the camera on board and is transmitted using a 2.4-GHz video emitter. The camera is pointing downwards and is located at the bottom of the helicopter coinciding with the center of mass. The ground station is composed of a video receiver of 2.4 GHz and a frame grabber card with capacity to process up to 30 frames per second (FPS), but in our applications the rate is around 15 FPS. The image processing codes are developed on C++ on a ground station computer to obtain the estimation of location and translational velocity of our flying machine. The data obtained from each image (or from two consecutive images for optical flow estimations) are transmitted via modem to the microprocessor embedded in the rotorcraft. Figure 6 shows a schematic representation of the avionics, communication and positioning systems that have been used.

8. Experimental Results

In this section, we present real-time experiments of the stabilization of a four-rotor rotorcraft using the method to estimate the position and orientation proposed in Section 3. In addition, the practical results obtained by applying the method described in Section 4 are also shown.

The calibration matrix process applying *the homogeneous transformation approach* has given the following intrinsic parameters matrix $\mathbf{K}$

$$\mathbf{K} = \begin{bmatrix} 430 & 2 & 226 \\ 0 & 438 & 161 \\ 0 & 0 & 1 \end{bmatrix}. \tag{35}$$

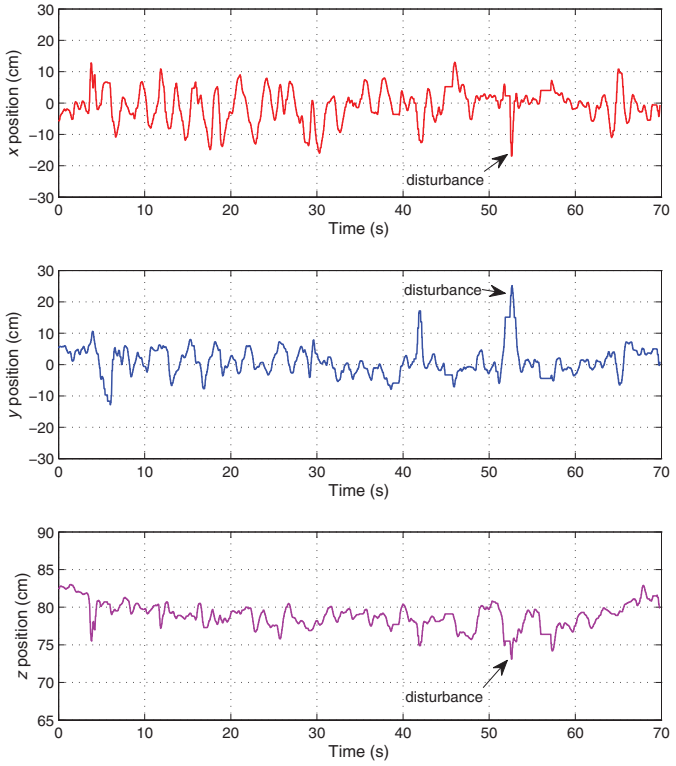


Fig. 7. (Colour online) Position estimation in the Cartesian space x, y, z.

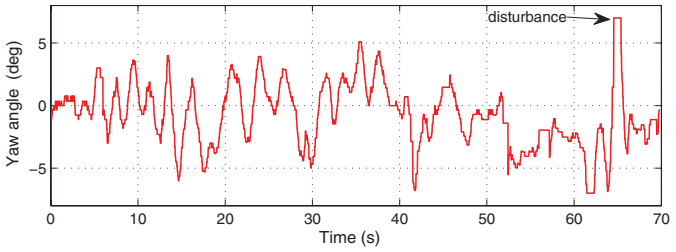


Fig. 8. (Colour online) Yaw angle of a rotorcraft computed with vision.

Practical results obtained with the combination of the *plane-based pose approach* and the *optical flow method* are presented in Figs. 7–9. The helicopter is stabilized at hover at an altitude of 80 cm. The horizontal displacement and the velocity are regulated around the origin. Cartesian coordinates (x, y, z), the vertical velocity $\dot{z}$ and yaw angle ψ are estimated using the geometrical approach described in Section 3, while the translational speeds $\dot{x}$ and $\dot{y}$ are obtained from the optical flow method presented in Section 4.[20] We can realize from Figs. 7 and 8 that the performance of the flying robot using the proposed computer vision-based control technique is satisfactory because the error position is bounded in an interval of ± 15 cm on both x- and y-axes, and the z-axis error is restricted to ± 5 cm. Furthermore, the measures corresponding to the yaw angle ψ are limited to ± 5 degrees of the error.

Figure 9 shows the control inputs applied to each motor. This control law does not produce saturation in the motors because the maximum PWM level supplied to them is around

https://doi.org/10.1017/S026357471200001X Published online by Cambridge University Press

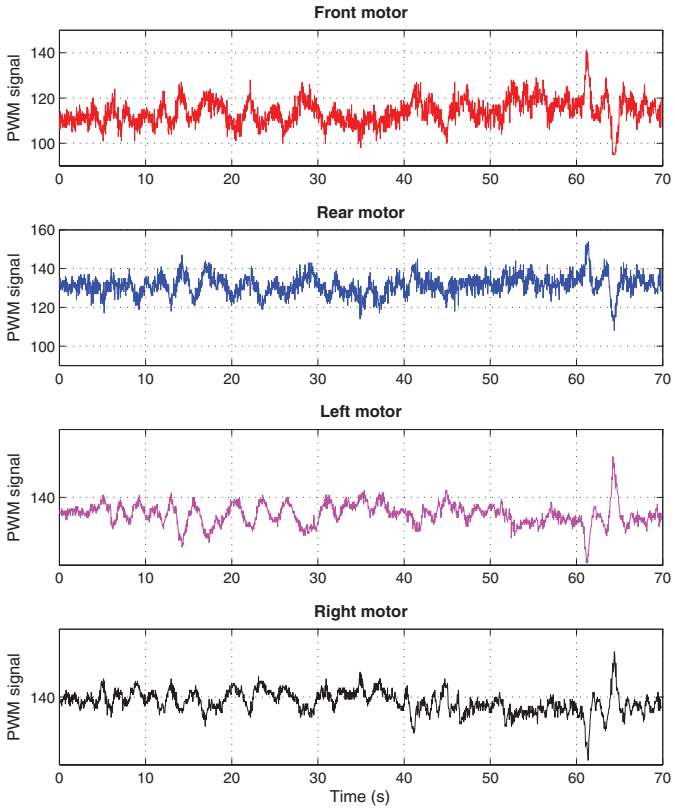


Fig. 9. (Colour online) Control inputs applied to a rotorcraft.

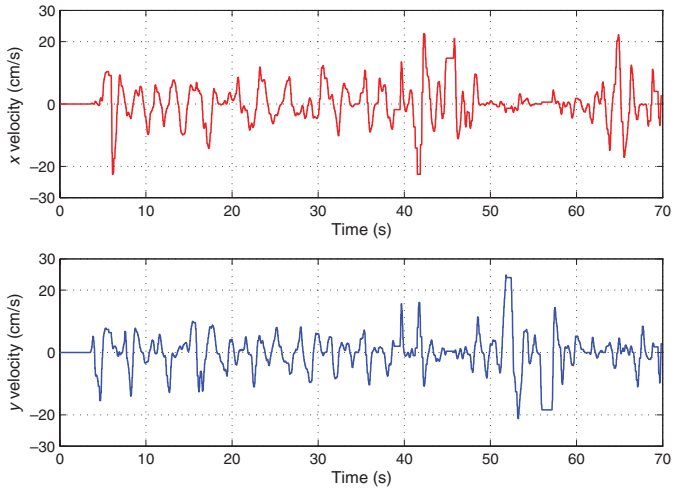


Fig. 10. (Colour online) Translational speed of the aircraft estimated from optical flow measurements.

160 units, which is 40 units below of the maximum allowed level for the driver.

Finally, we present the results obtained using only the optical flow approach under similar conditions of regulation for horizontal displacement and velocity. The estimated linear velocities $(\dot{x},\ \dot{y})$ and position $(x,\ y)$ are plotted in Figs. 10 and 11 respectively. We use this last part of the experiment to establish a comparison between this work and our previous work.[26] In the present case, we use only the translational velocity information estimated from the optical flow measurements and their respective position estimation obtained by simple integration of this translational velocity.

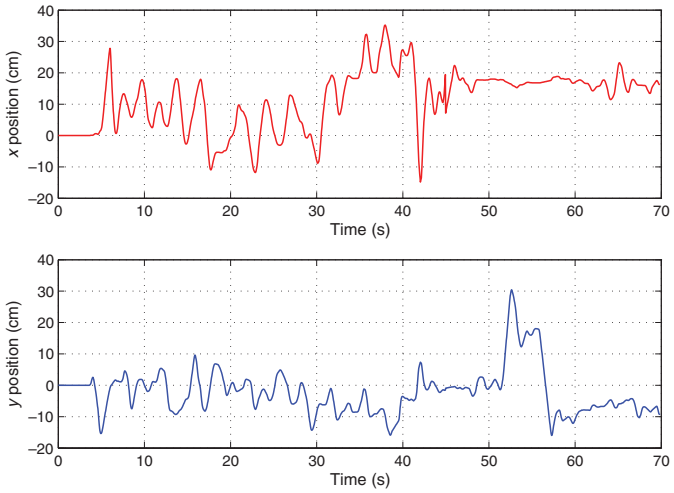


Fig. 11. (Colour online) Position estimated of the aircraft using optical flow integration.

In order to avoid drift, we reset the integrator output to zero every 100 s.

Figure 7 presents the position estimation using the geometric approach of Section 3, while Fig. 11 shows the position estimation using only the optical flow integration. The geometric approach in Fig. 7 performs better because it is robust with respect to disturbances, and the error with respect to the origin is smaller than in the case of Fig. 11.

A video showing an experimental performance can be seen at http://www.youtube.com/watch?v=zICSxP5m6fI.

9. Conclusions

In this paper we have presented a combination of a vision system with an IMU to compute the position and orientation in order to stabilize at hover a mini four-rotor rotorcraft. A methodology was developed and implemented in real time combining the *plane-based pose method* approach for estimating the position and the *pyramidal Lucas–Kanade* algorithm for estimating the translational speed with respect to a very well-known target. The target is composed of two pairs of two different symbols located in the vertices of a square. The monocular vision system was used to estimate the position in the Cartesian space $(x,\ y,\ z)$ and the yaw angle ψ. Measurements of the pitch and roll angles were obtained directly from an IMU operating at a higher rate as compared to the estimation algorithm performed by the vision system. The experimental tests showed that the proposed vision system-based control technique performs satisfactorily for stabilizing a mini-aerial vehicle at hover.

A second algorithm using a vision system together with IMU sensors is proposed to estimate attitude, position and velocity of the flying machine. This vision algorithm concerns the estimation of optical flow to measure the translational velocity of a rotorcraft and indirectly its position. Furthermore, we have used the angular speed information provided by the gyros to compensate the rotational speed component of the optical flow. However, oscillations in pitch and roll angles were still present and they introduced high-frequency noise in the optical flow estimation. Therefore, we have used a low pass filter on

https://doi.org/10.1017/S026357471200001X Published online by Cambridge University Press

optical flow to measure the translational velocity of a mini-helicopter. The real-time experiments have shown a satisfactory performance of a rotorcraft using the proposed computer-vision-based control technique.

The benefit of the technique proposed in this paper is that it provides a low cost sensor for estimating the whole state of a rotorcraft. This estimation is sufficient for stabilizing flying machine at hover.

References

1. C. H. Oertel, "Machine vision-based sensing for helicopter flight control," *Robotica* **18**(3), 299–303 (2000).
2. J. Shin, K. Nonami, D. Fujiwara and K. Hazawa, "Model-based optimal attitude and positioning control of small-scale unmanned helicopter," *Robotica* **23**(1), 51–63 (2005).
3. A. Wu, E. Johnson and A. Proctor, "Vision-Aided Inertial Navigation for Flight Control," *Proceedings of the AIAA Guidance, Navigation, and Control Conference and Exhibit*, San Francisco, CA (Aug. 15–18, 2005).
4. L. Mejias, S. Saripalli, P. Campoy and S. Sukhatme, "Visual servoing of an autonomous helicopter in Urban areas using feature tracking," *J. Field Robot.* **23**(3–4), 185–199 (2006).
5. Y. Yoshihata, K. Watanabe, Y. Iwatini and K. Hashimoto, "Multi-Camera Visual Servoing of a Micro Helicopter Under Occlusions," **In**: *Proceedings of IEEE International Conference on Intelligent Robots and Systems 2007*, pp. 2615–2620.
6. S. Saripalli, J. F. Montgomery and G. S. Sukhatme, "Visually-guided landing of an unmaned aerial vahicle," *IEEE Trans Robot. Autom.* **19**(3), 371–381 (2003).
7. K. Nordberg, P. Doherty, G. Farnebäck, P. E. Forssén, G. Granlund, A. Moe and J. Wiklund, "Vision for a UAV Helicopter," *Proceedings of the International Conference on Intelligent Robots and Systems (IROS) 2002*, Lausanne, Switzerland.
8. E. Altug, J. P. Ostrowski and C. J. Taylor, "Control of a quadrotor helicopter using dual camera visual feedback," *Int. J. Robot. Res.* **24**(5), 329–341 (May 2005).
9. O. Bourquardez, R. Mahony, N. Guenard, F. Chaumette, T. Hamel and L. Eck, "Image-based visual servo control of the translation kinematics of a quadrotor aerial vehicle," *IEEE Trans. Robot.* **25**(3), 743–749 (June 2009).
10. B. Herisse, F. Russotto, T. Hamel and R. Mahony, "Hovering Flight and Vertical Landing Control of a VTOL Unmanned Aerial Vehicle using Optical Flow, *Proceedings of the 2008 IEEE/RSJ International Conference on Intelligent Robots and Systems*, Nice, France (Sep. 22–26, 2008).
11. O. Amidi, T. Kanade and K. Fujita, "A visual odometer for autonomous helicopter flight," *J. Robot. Auton.* **28**, 185–193 (1999).
12. P. Corke, "An inertial and visual sensing system for a small autonomous helicopter," *J. Robot. Syst.* **18**, 43–51 (2004).
13. P. Corke, D. Strelow and S. Singh, Omnidirectional Visual Odometry for a Planetary Rover, *Proceedings of IEEE International Conference on Intelligent Robots and Systems (IROS) 2004*.
14. R. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*, 2nd ed. (Cambridge University Press, Cambridge, UK, 2004). ISBN 0521540518.
15. P. I. Corke, *Visual Control of Robots, High-Performance Visual Servoing*, 1st ed. (John Wiley, Hoboken, NJ, 1996). ISBN 0471969370.
16. H. Romero, Modelisation et Asservissement Visuel d'un Mini-hèlicoptére *Ph.D. Thesis* in French (Universié de Technologie de Compiègne France, Julio 2008).
17. P. Viola and M. Jones, "Rapid Object Detection Using a Boosted Cascade of Simple Features," *Proceedings of the IEEE Computer Society Conference on Computer Vision and Patter Recognition*. Vol. 1, 511–518 (2001).
18. P. Sturm, "Algorithms for plane-based pose estimation," **In**: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2000)*, June 13–15, Hilton Head, SC, pp. 706–711.
19. S. S. Beauchemin and J. L. Barron, "The computation of optical flow," *ACM Comput. Surv.* **27**, 433–467 (1995).
20. J. Y. Bouguet, "Pyramidal Implementation of the Lucas Kanade Feature Tracker," Technical report, Intel Corporation, Santa Clara, CA (1999).
21. W. E. Green, P. Y. Oh and G. L. Barrows, "Flying Insect Inspired Vision for Autonomous Aerial Robot Maneuvers in Near-Earth Environments," *Proceedings of IEEE International Conference on Robotics and Automation, ICRA* (2004).
22. P. Castillo, A. Dzul and R. Lozano, "Real-time stabilization and tracking of a four rotor mini rotorcraft," *IEEE Trans. Control Syst. Technol.* **12**(4) 510–516 (2004).
23. A. R. Tell, "Global stabilization and restricted tracking for multiple integrator with bounded controls," *Syst. Control Lett.* **18**, 165–171 (1992).
24. S. Salazar-Cruz, J. Escareño, D. Lara and R. Lozano, "Embedded control system for a four-rotor UAV," *Int. J. Adapt. Control Signal Process.* **21**(2–3), 189–204 (2007).
25. S. Merhav, *Aerospace Sensor Systems and Applications* (Springer-Verlag, New York, 1996). ISBN 0387946055.
26. H. Romero, S. Salazar and R. Lozano, "Real time stabilization of an eight-rotor UAV using optical flow," *IEEE Trans. Robot.* **25**(4), 809–817 (Aug. 2009).

https://doi.org/10.1017/S026357471200001X Published online by Cambridge University Press