Preprints of the 18th IFAC World Congress
Milano (Italy) August 28 - September 2, 2011

Vision Based Tracking for a Mini-Rotorcraft Using Vanishing Points ⋆

J.E. Gomez-Balderas * **P. Castillo** * **J.A. Guerrero** *
R. Lozano **

* *Heudiasyc UMR 6599 Université de Technologie de Compiégne, Centre de Recherches de Royallieu, 60200 France. e-mail: {jgomezba,castillo,jguerrer}@hds.utc.fr*
** *Heudiasyc UMR 6599, LAFMIA UMI CNRS 3175, Av. Instituto Politécnico Nacional No. 2508 San Pedro Zacatenco 07360 México, DF, México. e-mail: rlozano@hds.utc.fr*

Abstract: A vision based line tracking control strategy for a mini-rotorcraft is presented in this paper. The vision algorithm, employed to estimate the 3D position of the mini helicopter, is based on the vanishing points technique. Also, the dynamic model of this rotorcraft is presented in order to propose a nonlinear controller to navigate following a line. To validate the theoretical results, a real-time embedded control system has been developed. Experimental results have shown the good behavior of the proposed control vision scheme.

1. INTRODUCTION

Recently Unmanned Aerial Vehicles (UAVs) have become a standard platform for many applications, both civil and military. Civil applications range from patrol of the borders by aerial platforms, looking for survivors from shipwrecks, aircraft accidents, scientific research of any nature (environmental, atmospheric, archaeological, pollution etc), nuclear factory surveillance, surveillance of pipelines, buildings surveillance, wall surveillance etc. The family of distance sensors such as ultrasound sensors can only detect nearby obstacles. Sonar sensors and lasers scanner are usually too heavy for small platforms. A vision system can be used as a sensor to estimate the UAV position due to the great quantity of information coming from images.

An important problem of working with UAV is the autolocalization in unknown environments. Computer vision systems have been recently used as a solution to 3D position estimation of the mini-UAVs. Many visual tasks require recovering 3D information from sequences of images taking into account the world modeled as a projective space and determine how projective invariant information can be recovered from the images and used in robotics applications, see Luc et al. [1995]. The relationships amongst camera parameters, structures in 3D scenes and Vanishing Points (VP) have been established in Haralcik [1980]. Also, VP are widely employed for camera calibrations and recovery of rotational component of motion and real time pose estimation in an urban environment (ZuWhan [2006]). The usefulness of VP in motion analysis is the 3D orientation representation and the 3D translations between the camera and the scene. In addition, Wang [1990] and Guillou et al. [2000] have proposed different approaches to camera calibration based upon the use of vanishing lines. This techniques requires only a single view of a cube, because in the 3D space the parallel lines meet only at infinity, and a vanishing point being of course equivalent to the projection of a point at infinity. On the other hand, a method that not utilizes any 3D measurements of the circle or lines in many scenarios can be found in Wang et al [2008].

This paper focuses on estimating the 3D position of an UAV while it tracks a line painted on a wall with no marks. To solve this problem we use the vanishing points of the line to obtain the rotation matrix and the translation vector of a camera placed onboard of the helicopter and pointing forward. Additionally, another camera, placed also into the helicopter and pointing downward, is employed to calculate the optical flow and then translational velocity of the rotorcraft. The image information is thus introduced inside the closed-loop control of the vehicle to follow, autonomously, a line.

Several works dealing with the vision-based tracking can be found in the literature. We can quote here only some among them. Kamrani et al. [2007] have studied an online simulation approach in UAV path planning for tracking a single target using particle filter method. Also, a navigation strategy that employs the optical flow and inertial information to control a simulated helicopter is presented in Muratet et al. [2005]. Likewise, vision based structure detection and control design for tracking have been used for exploration in natural environments such as road following (Frew et al. [2004]), linear structures following (Rathinam et al. [2005]) and river tracking (Rathinam et al. [2007]).

The outline of this paper is as follows: the more important characteristics of the vanishing points technique are described in section 2. The 3D position algorithm is described in section 3 whilst in section 4 the dynamic model of the quadrotor is conceived using Newton-Euler method. The nonlinear control design is developed in section 5 and the experimental results are illustrated using some graphs in section 6. Finally, the conclusion is discussed in section 7.

⋆ This work was supported in part by the CNRS and CONACYT.

Copyright by the
International Federation of Automatic Control (IFAC)

2. VANISHING POINT DETECTION

Some objects in 3D space can be expressed in terms of parallelism, orthogonality and coplanarity, these constraints can be used to obtain three vanishing points in the image plane, that corresponds to three mutually orthogonal directions in space. Employing these vanishing points is possible to obtain intrinsic and extrinsic parameters of a camera.

A vanishing point, in a perspective geometry, is a single point where all the projected parallel segment lines in 3D world, which are projected to non-parallel lines in the image plane, converge. Rigorously speaking, the vanishing points can be found outside the field of view of the camera. If a set of lines is parallel to one of the three principal axes (x, y, or z) then, the point is called a principal vanishing point. In the 3D world, there exist three principal vanishing point, one for each axes named V_x, V_y and V_z. Computing one or more of these vanishing points from an image, helps to obtain an estimation of image depth, object dimension and 3D structure, see Ling [2008].

The vanish point approach could be defined as follows: the target should be located in front of the camera, i.e. in the field of view of the camera, and in order to simplify the analysis the target will be a rectangle with vertexes (A, B, C, D) in a 3D world. In the image plane this rectangle is represented as a 4-sided polygon with vertexes (a, b, c, d), see Figure 1. Thus, knowing the coordinates of each vertex in the image plane, it is possible to define: $\mathbf{l}^1$, $\mathbf{l}^2$, $\mathbf{m}^1$ and $\mathbf{m}^2$ like the projected lines formed by the pair of vertex (b, c), (a, d), (a, b) and (c, d), respectively. In addition, let V_x be a vanishing point obtained for intersecting lines $\mathbf{l}^1$ and $\mathbf{l}^2$ and similarly V_y will be the vanishing point obtained for intersecting lines $\mathbf{m}^1$ and $\mathbf{m}^2$, see Figure 1.

Once the intrinsic calibration parameters are known, it is possible to compute the external calibration parameters utilizing the vanishing points of any two axes of the reference of camera coordinate system. In mathematical form, it yields, $V_x = \mathbf{l}^1 \times \mathbf{l}^2, V_y = \mathbf{m}^1 \times \mathbf{m}^2$ where $V_x = \frac{[V_{xx}, V_{xy}, f]^T}{\sqrt{V_{xx}^2 + V_{xy}^2 + f^2}}$ and $V_y = \frac{[V_{yx}, V_{yy}, f]^T}{\sqrt{V_{yx}^2 + V_{yy}^2 + f^2}}$, and f is the focal length. Therefore, with the fact that V_x and V_y are orthogonal then, the vanishing point in the z-axis is $V_z = V_x \times V_y$. Thus, the rotation matrix which describes the rigid motion between the world and camera coordinate system becomes

$$R = [\, V_x \;\; V_y \;\; V_z \,] = \begin{bmatrix} \mathbf{i}^T \\ \mathbf{j}^T \\ \mathbf{k}^T \end{bmatrix} \tag{1}$$

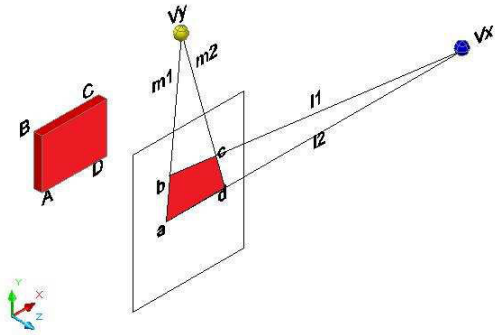

Fig. 1. Vanish points in a 4-sided polygon.

Obtaining the vanishing points is necessary to preprocess every image. This pre-treatment of the image involves a color segmentation process, to delete the image blur, and a boundaries extraction process to locate in the image plane four coplanar points which are the corners of a four-sided polygon.

Each vertex of the polygon has image coordinates given by (x_i, y_i), $\forall\ i = a, b, c, d$; which correspond in the 3D world to the unknown coordinates (X_i, Y_i, Z_i). In order to estimate these values, the POSIT approach is used. This technique uses the scaled orthographic projection p_i of the points (A, B, C, D) to calculate the translation vector $\mathbf{T}$. Moreover $\mathbf{T}$ represents the vector $\overrightarrow{OA}$ between the center of projection O and the reference point A which is the origin of the object coordinate frame of reference with coordinates (X_0, Y_0, Z_0). Since that a is the image point of A then, this vector is aligned with vector $\overrightarrow{Oa}$ and is equal to $\frac{Z_o}{f}\overrightarrow{Oa}$. Thus, the object pose is fully defined by i, j, x_a, y_a, and Z_o.

3. 3D POSITION ESTIMATION

The vision based 3D position estimation goal is to calculate the translation vector $\mathbf{T} = [T_x, T_y, T_z]^T$ of the frontal camera, using rotation matrix (1) obtained from the vanishing points.

The relationship between the coordinates of feature points A, B, C, D in the camera and object coordinate system can be expressed by:

$$\begin{bmatrix} X_i \\ Y_i \\ Z_i \end{bmatrix} = \begin{bmatrix} \mathbf{i}^T \\ \mathbf{j}^T \\ \mathbf{k}^T \end{bmatrix} \begin{bmatrix} U_i \\ V_i \\ W_i \end{bmatrix} + \begin{bmatrix} X_0 \\ Y_0 \\ Z_0 \end{bmatrix} \tag{2}$$

where $\mathbf{i}, \mathbf{j}$ and $\mathbf{k}$ are the unit vectors of the camera coordinate system expressed in the object coordinate system and (U_i, V_i, W_i) represent the coordinates of the points A, B, C, D.

Hence, the exact pose of the object in the reference point A can be expressed by:

$$x_0 = f\frac{X_0}{Z_0}; \qquad x_i = f\frac{X_i}{Z_i} \tag{3}$$

$$y_0 = f\frac{Y_0}{Z_0}; \qquad y_i = f\frac{Y_i}{Z_i} \tag{4}$$

where x_0 and y_0 define the image location of the origin of the object and x_i, y_i are the image coordinates of the points a, b, c, d. Introducing (2) into (3), it follows that

$$x_i = f\frac{\overrightarrow{AP_i} \cdot \mathbf{i} + X_0}{\overrightarrow{AP_i} \cdot \mathbf{k} + Z_0} = \frac{\overrightarrow{AP_i} \cdot \frac{f}{Z_0}\mathbf{i} + x_0}{\frac{1}{Z_0}\overrightarrow{AP_i} \cdot \mathbf{k} + 1}$$

where $P_i \in (B, C, D)$. Additionally, all points P_i need to satisfy the following constraints

$$\overrightarrow{AP_i} \cdot \mathbf{I} = x_i(1+\varepsilon_i) - x_0, \quad (5)$$

$$\overrightarrow{AP_i} \cdot \mathbf{J} = y_i(1+\varepsilon_i) - y_0, \quad (6)$$

with

$$\mathbf{I} = \frac{f}{Z_0}\mathbf{i}, \quad \mathbf{J} = \frac{f}{Z_0}\mathbf{j}, \quad \varepsilon_i = \frac{1}{Z_0}\overrightarrow{AP_i} \cdot \mathbf{k}$$

Therefore, to estimate the position of the camera the iterative pose algorithm, described by the following pseudocode, needs to be executed (see Dementhon [1995])

(1) $\varepsilon_{i(0)}$, $n = 1$.
(2) Beginning of the loop.
Solve for $\mathbf{i}, \mathbf{j}$ and Z_0 using equations (5) and (6).
When the object points are coplanar, the additional equality $\mathbf{i} \cdot \mathbf{j} = 0$, must be used, and two approximate poses are found.
(3) Compute $\varepsilon_{i(n)} = \frac{1}{Z_0}\overrightarrow{AP_i} \cdot \mathbf{k}$, with $\mathbf{k} = \mathbf{i} \times \mathbf{j}$. When the object points are coplanar, two sets of ε_i with opposite signs are found.
(4) If $|\varepsilon_{i(n)} - \varepsilon_{i(n-1)}| <$ *Threshold*, Exit
Else $n = n + 1$. Go to step 2.
(5) Exact pose = last approximate pose.

Using the previous algorithm, the pose (X_i, Y_i, Z_i) of the camera with respect to the line can be then computed.

4. NONLINEAR MODEL

The quadrotor helicopter is a useful prototype to learn about aerodynamic phenomena in flying machines that can hover. This vehicle is controlled only varying the angular speed of each of the four rotors. In addition, the front and the rear motors rotate counter-clockwise, while the other two motors rotate clockwise.

The general motion of the vehicle can be represented by a combination of translational and rotational motions, see Figure 2. The dynamics of the helicopter, under external forces applied to the center of mass and expressed in inertial frame, is designed by the Newton-Euler formalism

$$\dot{\xi} = \upsilon \quad (7)$$

$$m\dot{\upsilon} = \mathcal{R}T_{f_{\hat{\mathbf{k}}}} - mg\hat{\mathbf{k}} \quad (8)$$

$$\dot{\mathcal{R}} = \mathcal{R}\hat{\Omega} \quad (9)$$

$$\mathbf{I}\dot{\Omega} = -\Omega \times \mathbf{I}\Omega + \tau_A + \tau_{G_A} \quad (10)$$

where $\xi = (x, y, z)^T$ denotes the position of the centre of mass of the vehicle in the inertial frame $\mathcal{I}$, $\upsilon \in \mathcal{I}$ represents the linear velocity, Ω is the angular velocity vector in the body frame A. m denotes the total mass of the rotorcraft whilst $\mathbf{I} \in \mathbb{R}^{3\times3}$ the constant inertia matrix around the centre of mass. $\hat{\Omega}$ represents the skew-symmetric matrix of the vector Ω, $\mathcal{R}$ is the rotation matrix from A to $\mathcal{I}$. $T_{f_{\hat{\mathbf{k}}}}$ describes the main thrust, mg is the gravitational force applied to the vehicle and $\hat{\mathbf{k}}$ denotes the unit vector codirectional with the z-axis. $\tau_A = [\tau_\psi, \tau_\theta, \tau_\phi]^T$ expresses the generalized torques applied to the helicopter whilst τ_{G_A} the gyroscopic torques, see Castillo [2005].

Equations (7) and (8) represent the translational dynamics of the helicopter whilst (9) and (10) introduce the rotational dynamic.

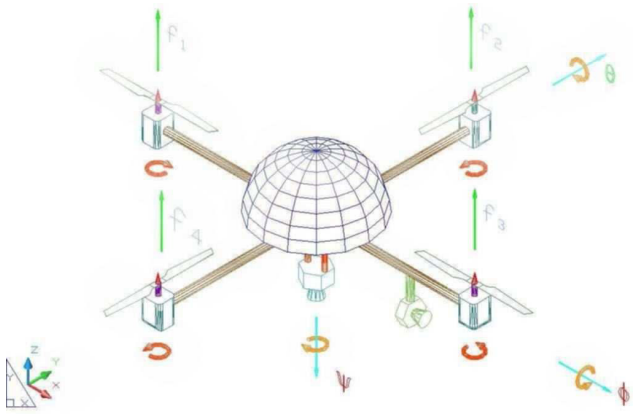

Fig. 2. Vehicle schematic for vertical flight mode

Notice that, the angular velocity vector, Ω, resolved in the body frame can be related to the generalized velocities $\dot{\eta}$, by the following standard kinematic relationship, $\Omega = W_\eta\dot{\eta}$, where W_η is the transformation matrix and $\eta = (\psi, \theta, \phi)$ denotes the orientation of the helicopter, more details see Goldstein [1980]. Thus, $\dot{\Omega} = W_\eta\ddot{\eta} + \dot{W}_\eta W_\eta^{-1}\Omega$. Introducing $\dot{\Omega}$ into (10), it yields

$$\mathbf{J}\ddot{\eta} = C(\eta, \dot{\eta})\dot{\eta} + \tau_{G_A} + \tau_A \quad (11)$$

where $\mathbf{J}$ is the inertia matrix for the full rotational kinetic energy of the helicopter expressed in terms of the generalized coordinates η and $C(\eta, \dot{\eta})\dot{\eta} = -I\dot{W}_\eta W_\eta^{-1}\Omega - \Omega \times \mathbf{I}\Omega$ represent the Coriolis matrix.

5. CONTROL SCHEME

The goal of this paper is to control the helicopter following a line using the vision system. The control law is based on saturation functions and obtained employing the Lyapunov analysis. In addition, the amplitudes of the saturation functions can be chosen in such a way that, after a finite time T', the angles lie in the interval -1 rad $\leq \eta \leq 1$ rad. Thus, the helicopter will be operated in quasi-stationary manoeuvres, and therefore, the coriolis and gyroscopics terms will tend to vanish. Hence, for further analysis, the following control input is designed

$$\tau_A = -C(\eta, \dot{\eta})\dot{\eta} - \tau_{G_A} + \tau_{\bar{A}} \quad (12)$$

where $\tau_{\bar{A}} = [\tau_{\bar{\psi}}, \tau_{\bar{\theta}}, \tau_{\bar{\phi}}]^T$ are the new control input vector.

Introducing (12) into (11) and rewriting the mathematical equations for the helicopter's dynamics, we get

$$m\ddot{x} = -T_f \sin\theta \quad (13)$$

$$m\ddot{y} = T_f \cos\theta \sin\phi \quad (14)$$

$$m\ddot{z} = T_f \cos\theta \cos\phi - mg \quad (15)$$

$$J_{zz}\ddot{\psi} = \tau_{\bar{\psi}} \quad (16)$$

$$J_{xx}\ddot{\theta} = \tau_{\bar{\theta}} \quad (17)$$

$$J_{yy}\ddot{\phi} = \tau_{\bar{\phi}} \quad (18)$$

Define $e_z = z - z_d$ and $e_\psi = \psi - \psi_d$ like the errors of the altitude and the yaw angle with respect to the desired constant values, z_d and ψ_d, respectively. Define now, the following control strategies

$$\frac{T_f}{m} = \frac{-a_{z_1}\dot{e}_z - a_{z_2}e_z + g}{\cos\theta\cos\phi} \tag{19}$$

$$\frac{\tau_{\bar{\psi}}}{J_{zz}} = -a_{\psi_1}\dot{e_\psi} - a_{\psi_2}e_\psi \tag{20}$$

where $a_{z_1}, a_{z_2}, a_{\psi_1}, a_{\psi_2}$ are positive constant. Introducing (19) and (20) into (15) and (16), respectively, it follows that exists a time T_a large enough such that, $\forall\ t > T_a$, $\dot{e}_z, \dot{e}_\psi, e_z, e_\psi \rightarrow 0$ and this implies that $z \rightarrow z_d$ and $\psi \rightarrow \psi_d$. Moreover, the control input for the lateral and longitudinal dynamic will be obtained using the Lyapunov analysis and the saturation functions, in such a way that the bounds can be chosen small enough to guarantee that $\cos\theta\cos\phi \neq 0$.

Rewriting the lateral and longitudinal dynamics when introducing (19), we have $\forall\ t > T_a$

$$\ddot{x} = -\frac{g\tan\theta}{\cos\phi} \qquad \ddot{y} = g\tan\phi$$
$$J_{xx}\ddot{\theta} = \tau_{\bar{\theta}} \qquad J_{yy}\ddot{\phi} = \tau_{\bar{\phi}}$$

The control objective is to follow a trajectory in the x axis while the z, ψ, ϕ and y are stabilized around to a desired constant values. In order to stabilize the lateral dynamics, we propose a nonlinear control law bounding every state, in such a way that the algorithm will guarantee an arbitrary bound for $\phi, \dot{\phi}, \dot{y}$ and y. To simplify further analysis, we will propose a very small upper bound on $|\phi|$ in such a way that the difference $\tan\phi - \phi$ is arbitrarily small. Thus, the lateral dynamics could be rewritten by

$$\ddot{y} \simeq g\phi$$
$$J_{yy}\ddot{\phi} = \tau_{\bar{\phi}}$$

Propose

$$\frac{\tau_{\bar{\phi}}}{J_{yy}} = -\sigma_{\phi_4}(k_{\phi_4}\dot{e}_y) - \sigma_{\phi_3}(k_{\phi_3}e_y) - \sigma_{\phi_2}(k_{\phi_2}\dot{e}_\phi) - \sigma_{\phi_1}(k_{\phi_1}e_\phi) \tag{21}$$

where $e_\phi = \phi - \phi_d$; and $e_y = y - y_d$; with ϕ_d, y_d are the constant desired values and $\sigma_{\phi_i}(\cdot)$ is a saturation function, see Sanahuja et al. [2009]. From Sanahuja et al. [2009], it follows that, $\dot{e}_\phi, \dot{e}_y, e_\phi, e_y \rightarrow 0$, and this implies that, $\phi \rightarrow \phi_d$ and $y \rightarrow y_d$. Choosing $\phi_d = 0$, it follows that exists a time T_b large enough such that $\forall t > T_b\ \cos\phi \approx 1$. In addition, the bound for the θ angle will be chosen such that $\tan\theta \approx \theta$.

For the longitudinal dynamics, let us propose $k_g e_x = x(t) - x_d(t)$ with $x_d(t) = b_0 + b_1 t + b_2 t^2$ is defined like the desired trajectory, $k_g = g/J_{xx}$ and b_i are constant. Thus $k_g\dot{e}_x = \dot{x} - \dot{x}_d$, $k_g\ddot{e}_x = \theta - \ddot{x}_d$, $k_g e_x^{(3)} = -g\dot{\theta}$, and

$$e_x^{(4)} = -\tau_{\bar{\theta}}$$

Again like for the lateral dynamics, we propose the following nonlinear control

$$\tau_{\bar{\theta}} = -\sigma_{\theta_4}(k_{\theta_4}\dot{e}_x) - \sigma_{\theta_3}(k_{\theta_3}e_x) + \sigma_{\theta_2}(k_{\theta_2}\dot{e}_\theta) + \sigma_{\theta_1}(k_{\theta_1}e_\theta) \tag{22}$$

The constant k_{ϕ_i} and k_{θ_i} are chosen like in Sanahuja et al. [2009] in such a way to ensure the convergence to zero. Thus, this implies that $e_x^{(4)}, e_x^{(3)}, \ddot{e}_x, \dot{e}_x, e_x \rightarrow 0$, and by consequence $\phi \rightarrow \ddot{x}_d$, $\dot{x} \rightarrow \dot{x}_d$ and $x \rightarrow x_d$.

6. EXPERIMENTAL RESULTS

In this section we present the results obtained by applying in real-time the algorithms proposed in previous sections.

6.1 Experimental Platform

The UAV is a mini helicopter with four rotors developed in our laboratory. The total weight of the vehicle is about 380 g, with a flight endurance of 10 minutes approximately. Theoretical results obtained were incorporated into an autopilot control system employing an architecture based on a 29 MHz *Rabbit* microcontroller, see Rabbit [2007]. We have built our own inertial measurement unit (IMU) using accelerometers, gyros and a compass to estimate the roll, pitch and yaw angles and also compute the angular rates. The IMU information is sent to the microcontroller, which also reads reference control inputs from user, through a serial wireless modem. The microcontroller subsequently combines this information to compute the control input and sends the control actions to the motors through an I2C serial port.

On the other hand, the vision system is composed of two cameras, *Web Philips SPC 1030NC*, placed orthogonally in the quadrotor, a camera is put downwards to process the optical flow to estimate the translational velocities while the other camera is looking forward, to get images of the line painted on a wall in front of the quadrotor vehicle. The images coming from the frontal camera are processed in a ground station in order to estimate the 3D position of the quadrotor using the vanishing point technique. In addition, the vision system periodically ($\sim 15 frames/sec$) determines the position and velocity of the rotorcraft and send this visual information to the vehicle via wireless communication. Figure 3 shows a block diagram of the basic architecture.

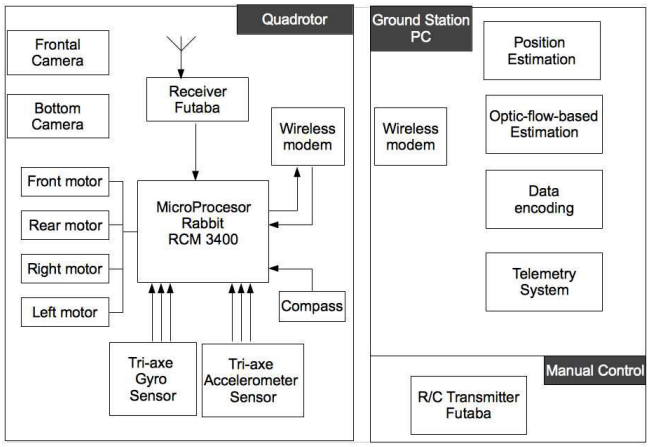

Fig. 3. Embedded Control Architecture

6.2 Experiments

The experiment has been devoted to stabilize the helicopter at $z_d = 80$ cm from the ground. Also, the helicopter should keep a desired distance of $y_d = 130$ cm from a wall while followed a line. Displacing the rotorcraft along the target line was necessary to have a variable displacement

in the range $[0, 230]$cm. Besides to estimate the velocity, the algorithm of Lucas-Kanade (Bouguet [1999]) was implemented and executed with the images obtained by the camera looking downwards. This algorithm finds a set of characteristic points in the ground and compare the displacement of this point, between two consecutive image frames. The optical flow can be generated by two kinds of observer motion: translational motion (F_t) and rotational motion (F_r), see Gomez-Balderas et al [2010].

The displacement over the $x-$axis was achieved by the information obtained from the frontal camera. This displacement has only two direction from left to right in the range $[0, 230]$ cm and vice-versa in the range $[230, 0]$ cm. In order to know what the direction of the displacement is, we have created a boolean flag to indicate the direction of displacement. This flag changes its value once that the quadrotor arrives to the range limit depending on the direction, indicating change of direction in the displacement of quadrotor.

Indeed, to find the 3D position the image is filtered to delete the blue and green channel leaving only the red channel in a RGB image format. In this new image a line detection algorithm is utilized. A search algorithm is then applied to find the coordinates of 4 corner points of the rectangle. These points are used to find the vanishing points Vx and Vy. The proposed vision algorithm processes images with a size of 640×480 pixels. All the vision algorithms are executed in a ground station composed by a PC with Intel Core 2 duo processor 2.10GHz using OpenCV libraries.

6.3 Results

The control gains were tuned in practice to obtain an acceptable system's response, *i.e.* they were selected to obtain a fast aircraft response but avoiding mechanical oscillations as much as possible. Theses values are shown in Table 1.

Parameter	Value		Parameter	Value
$a_{\psi 1}$	0.13		a_{z1}	2.5
$a_{\psi 2}$	2.0		a_{z2}	1.6
$k_{\theta 4}$	0.23		$k_{\theta 2}$	1.3
$k_{\theta 3}$	1.5		$k_{\theta 1}$	1.2
$k_{\phi 4}$	0.5		$k_{\phi 2}$	0.3
$k_{\phi 3}$	1		$k_{\phi 1}$	1.8
$\sigma_{\phi 4}$	35		$\sigma_{\theta 4}$	35
$\sigma_{\phi 3}$	30		$\sigma_{\theta 3}$	25
$\sigma_{\phi 2}$	20		$\sigma_{\theta 2}$	20
$\sigma_{\phi 1}$	18		$\sigma_{\theta 1}$	15

Table 1. Controller parameters values

Figure 4 (top) shows the displacement of the quadrotor over its $y-$axis. This measure estimation, corresponding to T_z value and obtained in centimeters, is obtained from the frontal camera. The small variations in the estimation are due to light changes and to the displacements of the helicopter. Observe in Figure 4 (middle) the quadrotor's trajectory over the $x-$axis, from the initial point (0cm) until the final point (230cm). Once that the final point has been reached, the helicopter changes its flight direction and the initial direction becomes the final direction. In same Figure (bottom) notices the altitude response of

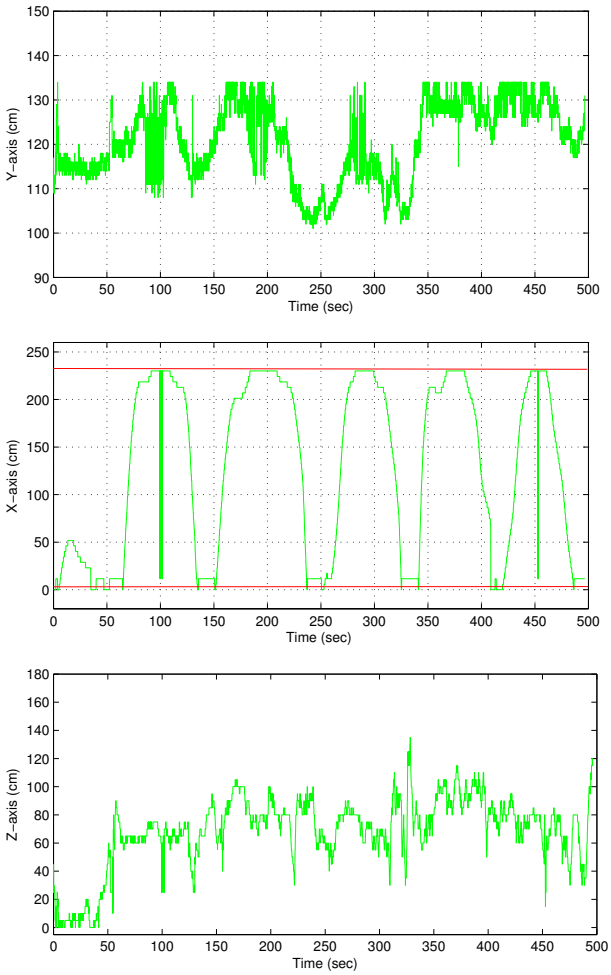

Fig. 4. X, Y and Z position of the mini-rotorcraft.

the vehicle. This estimation denotes the T_y value in the translational vector.

On the other hand, Figure 5 illustrates the velocity estimation when calculating the optical flow algorithm. These measurements are introduced into the closed-loop system to compute the control actions. Notice from these Figures the well performance of the proposed vision algorithms and the good states responses when applying the control strategy.

7. CONCLUSION

A vision based method which allows to estimate the 3D position of a mini quadrotor helicopter has been presented. The vision algorithm is based on the vanishing points method, to estimate the position of the rotorcraft, and on the Lucas-Kanade approach to compute the optical flow and by consequence the translational velocities. In addition, a control algorithm was developed in order to follow a trajectory. To validate the control and vision algorithms a quadrotor helicopter was used. This platform is composed of two cameras and an inertial measurement unit to compute in real-time the translational displacements (position and velocities) and also the rotational dynamic (orientation and angular rates). It has been shown that the proposed control scheme ensures the tracking of a visual

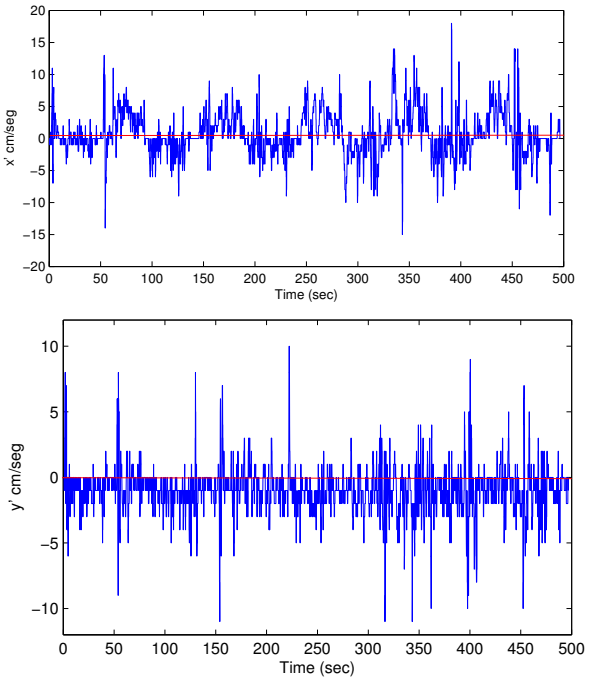

Fig. 5. $\dot{x}$ and $\dot{y}$ of the mini-rotorcraft.

reference. Future work includes the tracking of a nonlinear visual reference as well as the tracking of references in a structured environment.

REFERENCES

F. Kamrani and R. Ayani. Using on-line simulation for adaptive path planning of UAVs. *IEEE symposium on Distributed Simulation and Real-Time Applications*, 2007.

L. Muratet, S. Doncieux, et. al. A contribution to vision based autonomous helicopter flight in urban environments. *Robotics and Autonomous Systems* 50:195-209, 2005.

S. Salazar, S. Palomino, et. al. Trajectory tracking for a four rotor mini-rotorcraft. Proceedings of the 44th. *IEEE Conference on Decision and Control, and the European Control Conference*, 2005.

E. Frew, T. McGee, et. al. Vision-based road-following using a small autonomous aircraft. *IEEE Aerospace Conference Proceedings*, 5:3006-3015, 2004.

S. Rathinam, Z. Kim, et. al. Vision based following of locally linear structure using an unmanned aerial vehicle. *Proceedings of the IEEE conference on decision and control*, 2005.

S. Rathinam, P. Almeida, et. al. Autonomous searching and tracking of a river using an UAV. *Proceedings of the 2007 American Control Conference*, New York City, USA, July 11-13, 2007.

L. Robert, C. Zeller, et. al. Applications of non-metric vision to some visually guided robotics tasks. *Rapport de recherche n2584*, 1995.

R. M. Haralick. Using perspectives transformation in scene analysis. *Computer Graphics Image Processing*, 13:191-221, 1980.

K. ZuWhan. Geometry of vanishing Point and its Application to External and Realtime Pose Estimation. Research Reports, Institute of Transportation Studies, UC Berkeley. 2006.

L. Wang and W. Tsai. Computing camera parameters using vanishing line information from a rectangular parallelepiped. *Machine Vision and Applications*, (3):129-141, 1990.

E. Guillou, D. Meneveaux, et. al. Using vanishing points for camera calibration and coarse 3D reconstruction from a single image. *The Visual Computer*, 16:396-410, 2000.

G. Wang, J. Wu, et. al. Single view based pose estimation from circle or parallel lines. *Pattern Recognition Letters* 29:977-985, 2008.

Daniel F. Dementhon and Larry S. Davis. Model-Based Object Pose in 25 Lines of Code. *International Journal of Computer Vision* 15:123-141, 1995.

Ling-Ling Wang and Wen Tsai. Camera Calibration by Vanishing Lines for 3D Computer vision. *Pattern Recognition Letters* 29:977-985, 2008.

P. Castillo, R. Lozano and A. Dzul. Modelling and Control of Mini-Flying Machines (Advances in Industrial Control). *Springer 2005*

H. Goldstein. *Classical Mechanics*, Addison Wesley Series in Physics, Adison-Wesley, U.S.A., second edition, 1980.

G. Sanahuja, P. Castillo. Stabilization of n integrators in cascade with bounded input with experimental application to a VTOL laboratory system, *Int. J. Robust Nonlinear Control*.

Rabbit Semiconductors. *Dynamics C user manual*, http://www.rabbitsemiconductor.com/, 2007.

J.E. Gomez-Balderas, S. Salazar, A. Guerrero et L. Lozano. Vision Based Autonomous Hover of Miniature rotorcraft, *International Symposium on Unmanned Aerial Vehicles*, June 21-23, 2010.

J. Y. Bouguet. Pyramidal implementation of the Lucas Kanade feature tracker. *Technical report Intel Corporation*, 1999.

S. S. Beauchemin and J. L. Barron. The computation of optical flow. *ACM Computing Surveys*, Vol. 27, pp. 433-467, 1995.