



HAL
open science

Vision Based Tracking for a Quadrotor Using Vanishing Points

Jose-Ernesto Gomez-Balderas, Pedro Castillo, J. A. Guerrero, Rogelio Lozano

► **To cite this version:**

Jose-Ernesto Gomez-Balderas, Pedro Castillo, J. A. Guerrero, Rogelio Lozano. Vision Based Tracking for a Quadrotor Using Vanishing Points. International Conference of Unmanned Aircraft Systems (ICUAS 2011), May 2011, Denver, United States. hal-00999665

HAL Id: hal-00999665

<https://hal.science/hal-00999665>

Submitted on 3 Jun 2014

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

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

Vision Based Tracking for a Quadrotor Using Vanishing Points

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

the date of receipt and acceptance should be inserted later

Abstract In this paper, a vision based line tracking control strategy for mini-robotcraft is presented. In order to estimate the 3-D position of the mini-robotcraft over the trajectory a vanishing points technique is used. A dynamic model is derived employing the Newton-Euler approach and a nonlinear controller to stabilize, in closed-loop system, this mathematical model is proposed. To validate the theoretical results, a real-time embedded control system has been developed. The performance of the vision and control algorithms has been tested when the helicopter has tracked a line painted in a wall. The experimental results have shown the good behavior of the control laws.

Keywords Vision based control · Quadrotor · Tracking · Vanishing points · Nonlinear control

1 Introduction

Recently UAVs 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. Military applications includes Reconnaissance Surveillance and Target Acquisition (RSTA), surveillance for peacetime and combat Synthetic Aperture Radar (SAR), maritime operations (Naval fire support, over the horizon targeting, anti-ship missile defense, ship classification).

J. E Gomez-Balderas, Pedro Castillo, J. Guerrero and R. Lozano
Université de Technologie de Compiègne
Tel.: +33 3 44 23 44 23
Fax: +33 3 44 23 44 77
E-mail: (jgomezba,castillo,rlozano,jguerrer)@hds.utc.fr

To realize all these kind of applications we need to provide the UAV with high resolution sensors to collect data as well as to have a good estimation of the UAV position in the environment. 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.

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.

An important problem of working with UAV is the auto-localization 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 [6]. The relationships amongst camera parameters, structures in 3D scenes and Vanishing Points (VP) have been established in [7]. Also, VP are widely employed for camera calibrations and recovery of rotational component of motion and real time pose estimation in an urban environment [8]. The usefulness of VP in motion analysis is the 3D orientation representation and the 3D translations between the camera and the scene. In addition, [9] and [10] 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 [12].

Moreover, several works dealing with the vision-based tracking can be found in the literature. We can quote here only some among them. [1] have studied an on-line 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 [2]. Likewise, vision based structure detection and control design for tracking have been used for exploration in natural environments such as road following [3], linear structures following [4] and river tracking [5].

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

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 correspond to three mutually orthogonal directions in space. Using these vanishing points is possible to obtain intrinsic and extrinsic parameters of a camera.

In this article we propose to use the vanishing point detection technique to estimate the rotation matrix \mathbf{R} and translation vector \mathbf{T} of the camera by scaled orthographic projection and perspective projection. 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. Notice that 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 [11].

This approach will be described in the following and some assumptions are considered:

1. The target is located in front of the camera, i.e. in the field of view of the camera.
2. To simplify the analysis the target will be a rectangle with vertexes (A, B, C, D) in a 3D world.
3. In the image plane this rectangle is seen like a 4-sided polygon with vertexes (a, b, c, d) , see Figure 1.

Knowing the coordinates of each vertex in the image plane we define

- \vec{l}^1 as the projected lines formed by the pair of vertex (b, c) ,
- \vec{l}^2 , is the projected line formed by the pair of vertex (a, d) ,
- \vec{m}^1 is the projected line formed by the pair of vertex (a, b) and
- \vec{m}^2 is the projected line formed by the pair of vertex (c, d) .

Let V_x be a vanishing point obtained for intersecting lines \vec{l}^1 and \vec{l}^2 similarly, V_y is a vanishing point obtained for intersecting lines \vec{m}^1 and \vec{m}^2 , see Figure 1.

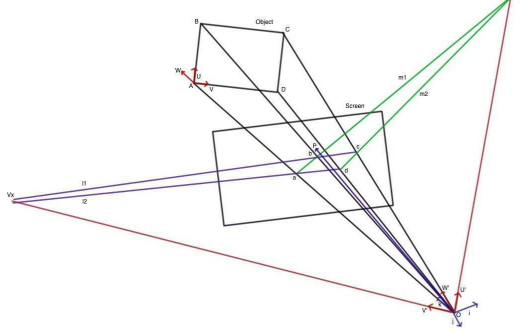


Fig. 1 Vanish points in a 4-sided polygon.

Once the intrinsic calibration parameters are known, in order to obtain the external calibration parameters all we need is the vanishing points corresponding to any two axes of the reference of camera coordinate system. The other axis is simply the cross-product of the two. In mathematical form, it yields,

$$\begin{aligned} V_x &= \vec{l}^1 \times \vec{l}^2 \\ V_y &= \vec{m}^1 \times \vec{m}^2 \end{aligned}$$

and 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 the focal length. Knowing that V_x and V_y are orthogonal then the vanishing point V_z over z axis of the camera is obtained by:

$$V_z = V_x \times V_y \quad (1)$$

The rotation matrix which describes the rigid motion between the world and camera coordinate system is:

$$\mathbf{R} = [V_x \ V_y \ V_z] \quad (2)$$

Observe that, in order to obtain the vanishing points, every image needs to be pre-processed. The pre-treatment of the image involves a color segmentation process to cancel 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.

Using the classic pinhole camera model, with its center of projection O , its image plane L at distance f from O its axes O_x and O_y pointing along the row and columns of the camera sensor, and O_z pointing along the optical axis. The unit vectors for these three axes are \mathbf{i}, \mathbf{j} and \mathbf{k} with coordinate system $R_c = (O, \mathbf{i}, \mathbf{j}, \mathbf{k})$. The 2D coordinate system associated with the screen plane can be denoted $R_s = (P, \mathbf{i}, \mathbf{j})$. For calibration operation we know f and the intersection of the optical axis with the image plane.

An object with characteristic points A, B, C, D is located in the field of view of the camera, in a plane M at a distance Z_A from the center of projection O , its coordinate system is R_o with a reference point A for the object, we can write $R_o = (A, U, V, W)$. Every point has a coordinate in the object coordinate frame $B = (U_B, V_B, W_B), C = (U_C, V_C, W_C)$ and $D = (U_D, V_D, W_D)$, with images points a, b, c, d . Each image point has a known image coordinate $p_i = (x_i, y_i) \forall i = A, B, C, D$ in the image plane. The coordinates (X_i, Y_i, Z_i) of the points A, B, C, D are unknown, because the pose of the object in the camera coordinate system is unknown. Note that the depths Z_i of different points are not very different from one other, and can all be set to the depth Z_A of the reference point A of the object. The coordinates (x'_i, y'_i) of the points p'_i , are the scaled orthographic projections of the feature points $P \in [B, C, D]$, called N_P . Knowing the height of the line ℓ , and the focal length f , we can calculate Z_A , using triangles properties.

Consider points A, N_P (see Figure 2) of the object and the plane M parallel to the image plane L , through the reference point A . The line of sight for N_P intersects plane M in M_P and N_P projects onto plane M in P . The vector \mathbf{AN}_P is equal to:

$$\mathbf{AN}_P = \mathbf{AM}_P + \mathbf{M}_P\mathbf{P} + \mathbf{PN}_P \quad (3)$$

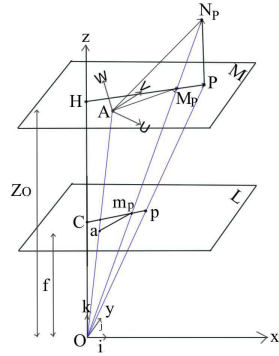


Fig. 2 Perspective projection m_P and scaled orthographic projection p for an object N_P and a reference point A .

The vector \mathbf{AM}_P and its image \mathbf{am}_P are proportional in the ratio $\frac{Z_A}{f}$. The vectors $\mathbf{M}_P\mathbf{P}$ and \mathbf{Cm}_P are also proportional in the two similar triangles Cm_PO and M_PPN_P , in a ratio equal to the ratio of the z coordinates of the other two corresponding vectors of these triangles \mathbf{PN}_P and \mathbf{OC} . This ratio is $\mathbf{PN}_P \cdot \frac{\mathbf{k}}{f}$. The sum of the three vectors can be expressed as:

$$\mathbf{AN}_P = \frac{Z_A}{f}\mathbf{am}_P + \mathbf{PN}_P \cdot \frac{\mathbf{k}}{f}\mathbf{Cm}_P + \mathbf{AN}_P \quad (4)$$

We take the dot product of this expression with the unit vector \mathbf{i} of the camera coordinate system. The dot product $\mathbf{PN}_P \cdot \mathbf{i} = 0$, the dot product $\mathbf{am}_P \cdot \mathbf{i}$ is the x -coordinate, $x_P - x_A$, of the vector $\mathbf{x}_A\mathbf{x}_P$. The dot product $\mathbf{Cm}_P \cdot \mathbf{i}$ is the coordinate x_P of \mathbf{Cm}_P . Defining $\beta_P = \frac{1}{Z_A}\mathbf{AP} \cdot R_3$ as the z -coordinate of \mathbf{AP} , and $R_3 = R_1 \times R_2$, where R_j is the j -th. row of rotation matrix \mathbf{R} , substituting we have:

$$\mathbf{AP} \cdot \frac{f}{Z_A}R_1 = x_P(1 + \beta_P) - x_A \quad (5)$$

Similarly, one obtains the following equation

$$\mathbf{AP} \cdot \frac{f}{Z_A}R_2 = y_P(1 + \beta_P) - y_A \quad (6)$$

taking the dot product of (4) with unit vector \mathbf{j} .

Now consider the points A, N_P , (see Figure 2) the projection P of N_P on the plane M , and its image p'_P with coordinates (x'_P, y'_P) . In scaled orthographic projection the vector \mathbf{AN}_P is the sum of two vectors

$$\mathbf{AN}_P = \mathbf{AP} + \mathbf{PN}_P \quad (7)$$

The vector \mathbf{AP} , and its image \mathbf{ap}_P , are proportional in the ratio $\frac{Z_A}{f}$. Consequently,

$$\mathbf{AN}_P = \frac{Z_A}{f}\mathbf{ap}_P + \mathbf{PN}_P \quad (8)$$

We take the dot product of this vector with unit vector \mathbf{i} of the camera coordinate system; the dot product $\mathbf{PN}_P \cdot \mathbf{i} = 0$, and the dot product $\mathbf{ap}_P \cdot \mathbf{i}$ is the x coordinate, $x'_P - x_A$, of the vector \mathbf{ap}_P . We obtain

$$\mathbf{AP} \cdot \frac{f}{Z_A}R_1 = x'_P - x_A \quad (9)$$

$$\mathbf{AP} \cdot \frac{f}{Z_A}R_2 = y'_P - y_A \quad (10)$$

Notice that the coordinates of p_P can be written as:

$$x'_P = x_P(1 + \beta_P) \quad (11)$$

$$y'_P = y_P(1 + \beta_P) \quad (12)$$

Knowing that A, B, C, D are coplanar, we must use $R_1 \cdot R_2 = 0$. The translation vector \mathbf{T} is aligned with vector OA , between the center of projection O and the reference point A . Since the image of A is the known image point a then \mathbf{T} is aligned with vector Oa and is equal to $(\frac{Z_A}{f})Oa$. Then we get:

$$X_A = \frac{Z_A}{f}x_A \quad (13)$$

$$Y_A = \frac{Z_A}{f}y_A \quad (14)$$

3 Nonlinear model

The quadrotor helicopter is a useful prototype for learning about aerodynamic phenomena in flying machines that can hover. Conventional helicopters modify the lift force by varying the collective pitch. These helicopters use a mechanical device known as swashplate to change the rotor blades pitch angle in a cyclic manner so as to obtain the pitch and roll control torques of the vehicle. The swashplate interconnects the servomechanisms and the blades pitch links. In contrast, the quad-rotor does not have a swashplate and has constant pitch blades. The quad-rotor is controlled only vary the angular speed of each of the four rotors. Additionally, the front and the rear motors rotate counter-clockwise, while the other two motors rotate clockwise. With this arrangement, gyroscopic effects and aerodynamic torques tend to cancel in trimmed flight.

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, T_f , is the sum of the individual thrusts of each motor. Forward pitch motion is obtained by increasing the speed of the rear motor, M_3 , while reducing the speed of the front motor, M_1 . Likewise, roll motion is obtained using the lateral motors. Yaw motion is obtained by increasing the torque of the front and rear motors (τ_{M_1} and τ_{M_3} respectively) while decreasing the torque of the lateral motors (τ_{M_2} and τ_{M_4} respectively). These motions can be accomplished while keeping the total thrust constant.

The general motion of the vehicle can be represented by a combination of translational and rotational motions. Thus, let us consider an inertial frame \mathcal{I} and a body frame \mathcal{A} . The center of mass of the helicopter and the body frame origin are assumed to coincide. Using Euler angles parametrization, the airframe orientation in space is given by a rotation \mathcal{R} from \mathcal{A} to \mathcal{I} , where $\mathcal{R} \in SO(3)$ is the rotation matrix. Then, the dynamics of the vehicle, under external forces applied to the center of mass and expressed in inertial frame,

is designed by the Newton-Euler formalism

$$\begin{aligned}\dot{\xi} &= v \\ m\dot{v} &= F \\ \dot{\mathcal{R}} &= \mathcal{R}\hat{\Omega} \\ \mathbf{I}\dot{\Omega} &= -\Omega \times \mathbf{I}\Omega + \tau\end{aligned}\tag{15}$$

where $\xi = (x, y, z)^T \in \mathcal{I}$ denotes the position of the centre of mass of the vehicle, $v \in \mathcal{I}$ represents the linear velocity, $\Omega \in \mathcal{A}$ is the angular velocity vector. m denotes the total mass of the rotorcraft whilst $\mathbf{I} \in \mathbb{R}^{3 \times 3}$ the constant inertia matrix around the centre of mass. $\hat{\Omega}$ represents the skew-symmetric matrix of the vector Ω . $F \in \mathcal{I}$ contains the vector of the principal non-conservative forces applied to the helicopter including thrusts T_f and drag terms associated with the rotors. $\tau \in \mathcal{A}$ is derived from differential thrust associated with pairs of rotors along with aerodynamic effects and gyroscopic effects.

Translational and gravitational forces

The only forces acting in the corps (and applied in the z -direction) are given by the translational force T_f and the gravitational force g . From Figure 3, notice that $T_f = \sum_{i=1}^4 f_i$, where the lift f_i generated by a rotor, z -direction and in free air, can be modeled as $f_i = k\omega_i^2$, with $k > 0$ is a constant and ω_i is the angular speed of the i -th motor. On the other hand, the gravitational force applied to the vehicle is $f_g = -mg\hat{\mathbf{k}}$, where $\hat{\mathbf{k}}$ denotes the unit vector codirectional with the z -axe. Therefore, f yields

$$F = \mathcal{R}T_{f_{\hat{\mathbf{k}}}} + f_g.\tag{16}$$

where $T_{f_{\hat{\mathbf{k}}}}$ represents the main thrust corresponding to the $\hat{\mathbf{k}}$ unit vector codirectional with the z -axe.

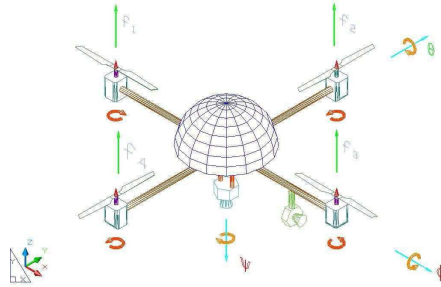


Fig. 3 Vehicle schematic for vertical flight mode

Torques

Due to the rigid rotor constraint the dynamics of each rotor disk around its axis of rotation can be treated as a decoupled system in the generalized variable ω_i denoting angular velocity of a rotor around its axis. The torque exerted by each electrical motor is denoted τ_{M_i} . The motor torque is opposed by an aerodynamic drag, τ_{drag} .

For definition, the aerodynamic drag is given by $\tau_{drag} = \frac{1}{2}\rho Av^2$, where ρ is the density of the air, A is the frontal area of the moving shape, and v is its velocity relative to the air. Remember that, in magnitude the angular velocity ω is equal to the linear velocity v divided by the radius of rotation r , i.e., $\omega = \frac{v}{r}$. Thus, the aerodynamic drag yields $\tau_{drag} = k_{drag} \omega^2$, where $k_{drag} > 0$ is a constant depending on the density of air, the radius, shape of the blade and others factors.

Using Newton's second law we have

$$I_M \dot{\omega}_i = \tau_{M_i} - \tau_{drag},$$

where I_M is the moment of inertia of the i -rotor around its axis. For quasi-stationary manoeuvres, $\omega \approx cte$, then

$$\tau_{M_i} \approx k_\tau \omega_i^2.$$

where $k_\tau > 0$ is a constant for steady state. Thus, from the characteristics of the quad-rotor, it follows that

$$\tau_A = \begin{bmatrix} \tau_\psi = \sum_{i=1}^4 \tau_{M_i} \\ \tau_\theta = (f_2 - f_4)\ell \\ \tau_\phi = (f_3 - f_1)\ell \end{bmatrix}$$

where τ_A represents the generalized torques applied to the helicopter, $\tau_\psi, \tau_\theta, \tau_\phi$ are the yawing, pitching, and rolling moment, respectively. ℓ is the distance between the motors and the center of mass of the vehicle. Rewriting the above,

$$\begin{aligned} \tau_\psi &= k_\tau (\omega_1^2 + \omega_3^2 - \omega_2^2 - \omega_4^2) \\ \tau_\theta &= \ell k (\omega_3^2 - \omega_1^2) \\ \tau_\phi &= \ell k (\omega_2^2 - \omega_4^2) \end{aligned}$$

The rotation axis of the i -rotor is moving with the angular velocity of the airframe, ω_f . This leads to the following gyroscopic torques applied to the airframe

$$\tau_{G_A} = -(\omega_f \times \hat{\mathbf{k}}) \sum_{i=1}^4 I_M \omega_i$$

Consequently, the total torque τ could be expressed by

$$\tau = \tau_A + \tau_{G_A} \quad (17)$$

Introducing (16) and (17) into (15), it follows that

$$\dot{\xi} = v \quad (18)$$

$$m\dot{v} = \mathcal{R}T_{f_k} - mg\hat{\mathbf{k}} \quad (19)$$

$$\dot{\mathcal{R}} = \mathcal{R}\hat{\omega} \quad (20)$$

$$\mathbf{I}\dot{\Omega} = -\Omega \times \mathbf{I}\Omega + \tau_A + \tau_{G_A} \quad (21)$$

Equations (18) and (19) represent the translational dynamics of the helicopter whilst (20) and (21) introduce the rotational dynamic.

Notice that, the angular velocity vector, Ω , resolved in the body frame can be related to the generalized velocities $\dot{\eta}$ (in the region where the Euler angles are valid), 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 [17]. Thus, $\dot{\Omega}$ can be expressed by

$$\dot{\Omega} = W_\eta\ddot{\eta} + \dot{W}_\eta W_\eta^{-1}\Omega$$

Introducing the above into (21), it yields

$$\mathbf{J}\ddot{\eta} = C(\eta, \dot{\eta})\dot{\eta} + \tau_{G_A} + \tau_A \quad (22)$$

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.

4 Control scheme

The goal of this paper is to control the helicopter following a line using the vision system. The control law is based in 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 lies in the interval $-1 \text{ rad} \leq \eta \leq 1 \text{ rad}$. Thus, the helicopter will be operated in quasi-stationary manoeuvres, and therefore, the coriolis and gyroscopics terms tend to vanish. Hence, for further analysis, the following control input is designed

$$\tau_A = -C(\eta, \dot{\eta})\dot{\eta} - \tau_{G_A} + \tau_{\bar{A}} \quad (23)$$

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

Introducing (23) into (22) and rewriting the mathematical equations for the helicopter's dynamics, we get

$$m\ddot{x} = -T_f \sin \theta \quad (24)$$

$$m\ddot{y} = T_f \cos \theta \sin \phi \quad (25)$$

$$m\ddot{z} = T_f \cos \theta \cos \phi - mg \quad (26)$$

$$J_{zz}\ddot{\psi} = \tau_{\bar{\psi}} \quad (27)$$

$$J_{xx}\ddot{\theta} = \tau_{\bar{\theta}} \quad (28)$$

$$J_{yy}\ddot{\phi} = \tau_{\bar{\phi}} \quad (29)$$

Define

$$\begin{aligned} e_z &= z - z_d \\ e_\psi &= \psi - \psi_d \end{aligned}$$

like the errors of the altitude and the yaw angle with respect to the desired values, z_d and ψ_d , respectively. In addition, z_d and ψ_d are constant, so that, $\dot{e}_z = \dot{z}$ and $\dot{e}_\psi = \dot{\psi}$. Define now, the following control strategies

$$\frac{T_f}{m} = \frac{-a_{z1}\dot{e}_z - a_{z2}e_z + g}{\cos\theta \cos\phi} \quad (30)$$

$$\frac{\tau_{\bar{\psi}}}{J_{zz}} = -a_{\psi1}\dot{e}_\psi - a_{\psi2}e_\psi \quad (31)$$

where $a_{z1}, a_{z2}, a_{\psi1}, a_{\psi2}$ are positive constant. Introducing (30) and (31) into (26) and (27), respectively, it follows that $\exists T_a$ 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 (30), we have $\forall t > T_a$

$$\begin{aligned} \ddot{x} &= -\frac{g \tan\theta}{\cos\phi} \\ J_{xx}\ddot{\theta} &= \tau_{\bar{\theta}} \\ \ddot{y} &= g \tan\phi \\ J_{yy}\ddot{\phi} &= \tau_{\bar{\phi}} \end{aligned} \quad (32)$$

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 further simplify the 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

$$\begin{aligned} \ddot{y} &\simeq g\phi \\ J_{yy}\ddot{\phi} &= \tau_{\bar{\phi}} \end{aligned}$$

Propose

$$\frac{\tau_{\bar{\phi}}}{J_{yy}} = -\sigma_{\phi_4}(k_{\phi_4}\dot{e}_y) - \sigma_{\phi_3}(k_{\phi_3}e_y) - \sigma_{\phi_2}(k_{\phi_2}\dot{\phi}) - \sigma_{\phi_1}(k_{\phi_1}\phi) \quad (33)$$

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 [18]. From [18], 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 $\exists T_b$ large enough such that $\forall t > T_b \cos\phi \approx 0$. In addition, the bound for the θ angle will be chosen such that $\tan\theta \approx \theta$.

For the longitudinal dynamics, let us propose $k_g e_x = x - x_d$ 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 $k_g \dot{e}_x = \dot{x} - \dot{x}_d$, $k_g \ddot{e}_x = \ddot{x} - \ddot{x}_d$, $k_g e_x^{(3)} = \dot{\theta}$, and

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

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

$$J_{xx} \tau_{\bar{\theta}} = \sigma_{\theta_4} (k_{\theta_4} \dot{e}_x) + \sigma_{\theta_3} (k_{\theta_3} e_x) - \sigma_{\theta_2} (k_{\theta_2} \dot{e}_\theta) - \sigma_{\theta_1} (k_{\theta_1} e_\theta) \quad (34)$$

The constant k_{ϕ_i} and k_{θ_i} are chosen like in [18] in such a way to ensure the convergence to zero. Thus, this implies that $e_x^{(4)}, e_x^{(3)}, \dot{e}_x, e_x \rightarrow 0$, and by consequence $\phi \rightarrow \ddot{x}_d$, $\dot{x} \rightarrow \dot{x}_d$ and $x \rightarrow x_d$.

5 Experimental Result

In this section we present the results obtained by applying the control strategy proposed in previous sections. A mini-quadrotor UAV is using a X4 type .

5.1 Platform

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 using an architecture based on a 29 MHz Rabbit microcontroller with 512 Kb Flash and 512Kb RAM. These microcontrollers are capable of handling floating point operations and multitasking processing virtually due to the enhancement compiler Dynamic C [16]. We have built our own inertial measurement unit (IMU) using accelerometers, gyros and a compass to obtain the roll, pitch and yaw angles and angular rates. The IMU information is sent to the microcontroller which also reads reference control inputs from an operator through a serial wireless modem. The microcontroller subsequently combines this information to compute the control input and sends the control corrections to the motors through a I2C serial port. Hence, the brushless motor speed controller or booster is handled by the I2C port of the microcontroller. Visual sensor is composed of two cameras Web placed orthogonally in the quadrotor, a camera is putting downwards to process an optical flow algorithm to obtain the velocity estimation, the other camera was placed in front of the X4 looking forward, to get images of a line painted on a wall in front of the quadrotor vehicle. The images coming from the frontal camera is used to estimate the 3D position of the quadrotor using vanishing points. Figure 4 shows a block diagram of the basic architecture.

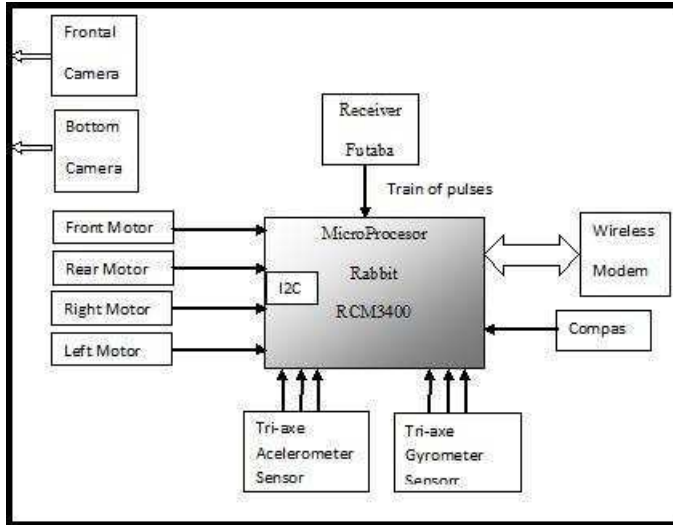


Fig. 4 Embedded Control Architecture

5.2 Experiments

To test the proposed algorithm was necessary to initialize the desired position values, in rotorcraft coordinates the values are assigned in this manner: $y_d = 130\text{cm}$. That is the distance desired between the quadrotor and the target, this value remains constant in all the experiment. The desired altitude was set $z_d = 80\text{cm}$. To displace the rotorcraft along the target line was necessary to have a variable displacement in the range $[0, 230]\text{cm}$. To estimate the velocity, the algorithm of Lucas-Kanade [15] was implemented in the camera mounted under the rotorcraft 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 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 $[0, 230]$ that was the initial constraint and the displacement from right to left $[230, 0]$. To know what the direction of the displacement is, we create 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, see Figure 5. These points are used to find the vanishing points V_x and V_y .



Fig. 5 Image with the four corner points.

The Optical flow can be generated by two kinds of observer motion: translational motion (F_t) and rotational motion (F_r). Let us assume that the camera is moving with translational velocity v and angular velocity ω while viewing an object at distance d and offset β from the direction of travel. Thus, the optical flow (OF) can be mathematically expressed as follows:

$$OF = \frac{v}{d} \sin \beta - \omega \quad (35)$$

The maximal optical flow is obviously generated when the plane that contains the features is perpendicular to translational motion direction ($\beta = 90^\circ$) [14]. Then, the velocity can be estimated of (35) as follows

$$v = \frac{(OF + \omega)}{\sin \beta} d \quad (36)$$

Notice that singularities in the above equation appear when $\beta = 0^\circ$. Nevertheless in our case the roll and pitch angles are very close to zero which implies that $\beta = 90^\circ \pm 3^\circ$ as a maximal interval when the mini-helicopter is appropriately stabilized at hover. The measurement of the angular speed ω is obtained by using the gyro information on each axis. An altitude stabilization algorithm in closed loop is used to keep the distance d constant and equal to some desired value.

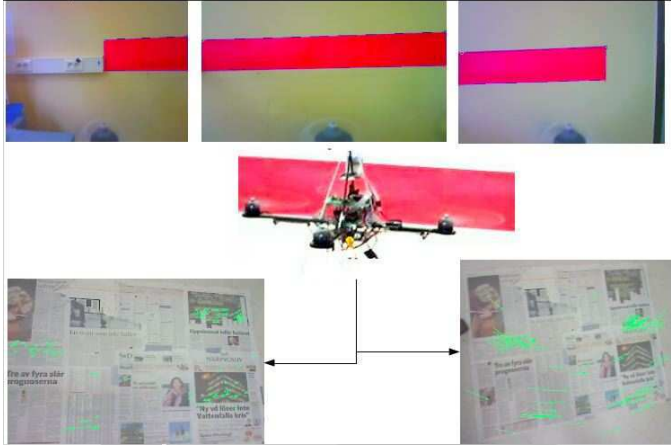


Fig. 6 Quadrotor helicopter following a line. Top images are used for the position algorithm while bottom images are employed in the velocity estimation.

In addition, 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.

5.3 Results

Figure 7 shows the displacement of quadrotor over its y -axis, this information is obtained from the frontal camera and it is the value of T_z in camera coordinates, the third element in translation vector \mathbf{T} of this camera. The distance obtained from the visual algorithm is shown in centimeters, it is possible to observe some variation in the estimation due to light changes over the displacement nonetheless the control law guarantees stable flight.

In Figure 8 is possible to see the quadrotor's trajectory over its x -axis which is parallel to the line, from the initial point (0cm) until the final point (230cm). Once that the final point has been reached, the translational displacement direction is changed by exchanging the initial and final points values. The distance covered is equal to the line length that is 230cm. The lateral position of the quadrotor was obtained from the T_x value in translation vector \mathbf{T} , obtained from frontal camera.

The figure 9 shows the quadrotor altitude T_y in translation vector \mathbf{T} estimated by the frontal camera, the desired altitude was set to 80cm. from ground.

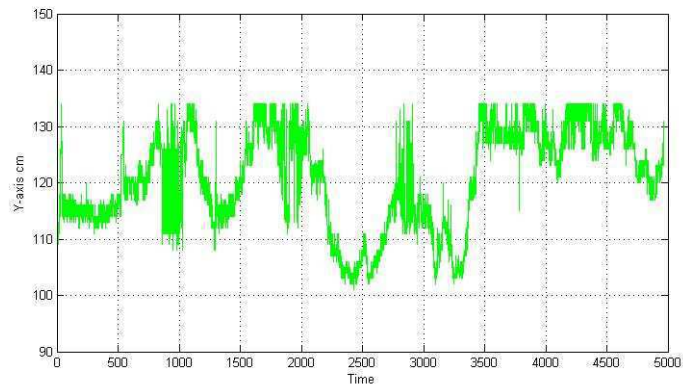


Fig. 7 Y position of mini-rotorcraft

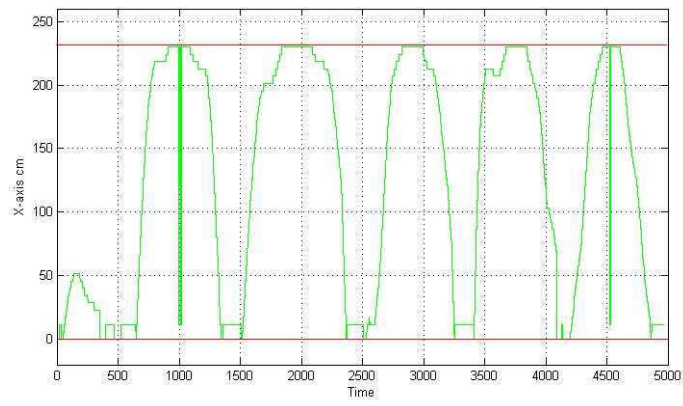


Fig. 8 X position of mini-rotorcraft

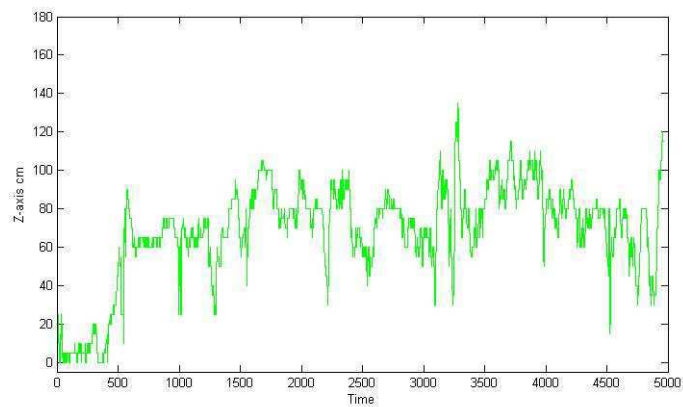


Fig. 9 Z position of mini-rotorcraft

Figures 10 and 11 show the velocity estimation of the quadrotor, obtained by using an optical flow algorithm applied on the images coming from downward camera. Both figures show quadrotor displacement performing line tracking experiments. The velocity estimation \dot{y} is the displacement over y -axis quadrotor presented in Figure 10. Figure 11 shows the velocity estimation, named \dot{x} , over x -axis of the quadrotor.

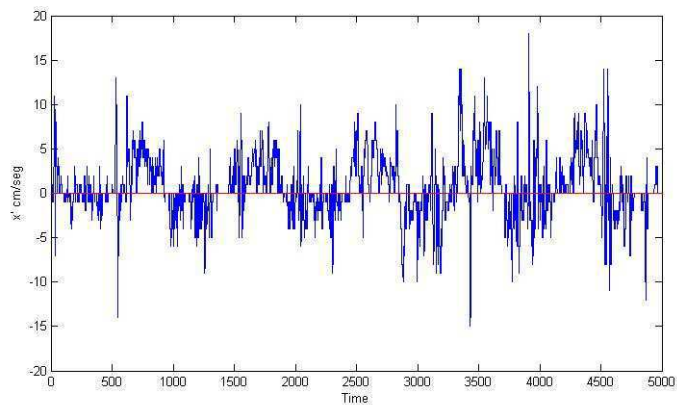


Fig. 10 \dot{x} velocity of the mini-robotcraft

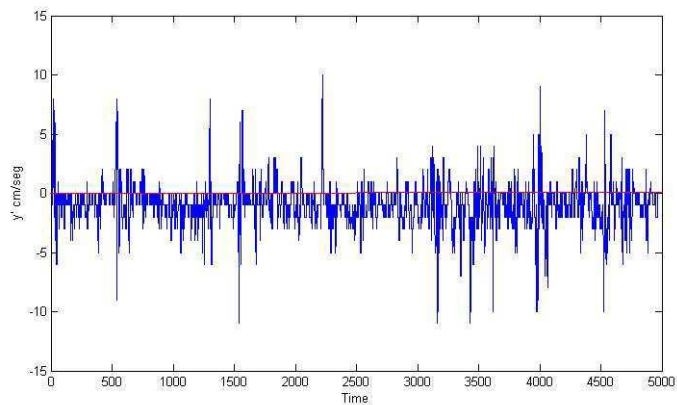


Fig. 11 \dot{y} velocity of the mini-robotcraft

6 Conclusions and future works

A vision based method to estimate the 3D position of a mini quadrotor has been presented. The proposed vision system is based on the vanishing points method. Similarly the velocity estimation is obtained using an optic flow algorithm. The estimated position and velocity information obtained from the computer vision system is combined with the angular rates and angles of the inertial measurement unit to compute the control inputs. It has been shown that the proposed control scheme ensures the tracking of a visual reference. Future works include tracking a non-linear visual reference, as well as tracking of reference in a structured environment.

References

1. 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.
2. 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.
3. E. Frew, T. McGee, et. al. Vision-based road-following using a small autonomous aircraft. *IEEE Aerospace Conference Proceedings*, 5:3006-3015, 2004.
4. 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.
5. 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.
6. L. Robert, C. Zeller, et. al. Applications of non-metric vision to some visually guided robotics tasks. *Rapport de recherche n2584*, 1995.
7. R. M. Haralick. Using perspectives transformation in scene analysis. *Computer Graphics Image Processing*, 13:191-221, 1980.
8. K. ZuWhan. Geometry of vanishing Point and its Application to External and Realtime Pose Estimation. Research Reports, Institute of Transportation Studies, UC Berkeley. 2006.
9. L. Wang and W. Tsai. Computing camera parameters using vanishing line information from a rectangular parallelepiped. *Machine Vision and Applications*, (3):129-141, 1990.
10. E. Guillou, D. Meneveau, et. al. Using vanishing points for camera calibration and coarse 3D reconstruction from a single image. *The Visual Computer*, 16:396-410, 2000.
11. Ling-Ling Wang and Wen Tsai. Camera Calibration by Vanishing Lines for 3-D Computer vision. *Pattern Recognition Letters* 29:977-985, 2008.
12. G. Wang, J. Wu, et. al. Single view based pose estimation from circle or parallel lines. *IEEE Transactions on pattern analysis and machine intelligence* 13:4, 1991.
13. 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.
14. S. S. Beauchemin and J. L. Barron, The computation of optical flow. In *ACM Computing Surveys*, Vol. 27, pp. 433-467, 1995.
15. J. Y. Bouguet, "Pyramidal implementation of the Lucas Kanade feature tracker". In *Technical report Intel Corporation*, 1999.
16. Rabbit Semiconductors, *Dynamics C user manual*, <http://www.rabbitsemiconductor.com/>, 2007.
17. H. Goldstein, *Classical Mechanics*, Addison Wesley Series in Physics, Addison-Wesley, U.S.A., second edition, 1980.

-
18. G. Sanahuja, P. Castillo and A. Sanchez, "Stabilization of n integrators in cascade with bounded input with experimental application to a VTOL laboratory system", *Int. J. Robust Nonlinear Control*, Published online in Wiley InterScience (www.interscience.wiley.com). DOI : 10.1002/rnc.1494, (2009).