Proceedings of the
44th IEEE Conference on Decision and Control, and
the European Control Conference 2005
Seville, Spain, December 12-15, 2005

Visual Tracking Control of Aerial Robotic Systems with Adaptive Depth Estimation

Najib Metni, Tarek Hamel, and François Derkx

Abstract— **This paper describes a visual tracking control law of an Unmanned Aerial Vehicle (UAV) for monitoring of structures and maintenance of bridges. It presents a control law based on computer vision for quasi-stationary flights above a planar target. The first part of the UAV's mission is the navigation from an initial position to a final position to define a desired trajectory in an unknown 3D environment. The proposed method uses the homography matrix computed from the visual information and derives, using backstepping techniques, an adaptive nonlinear tracking control law allowing the effective tracking and depth estimation. The depth represents the desired distance separating the camera from the target.**

Index Terms— **Aerial robots, Vision based navigation, Guidance and control, parameter identification.**

I. INTRODUCTION

Visual servoing techniques concern the problem of using a camera to provide information of position and attitude of a robotic system as well as to help tracking a certain predetermined trajectory. Micro aerial vehicles (called also Unmanned Aerial Vehicles) are often required to execute complex tasks (such as inspection or long time hovering) in unknown environments. To enable autonomous detection and navigation of these UAV, almost all control theories are built around a vision system by using visual servoing as a control method [1], [2]. A typical vision system will include a camera, an Inertial Navigation System (INS) in order to compute the attitude, orientation and velocity of the vehicle. Many vision applications involving mobile robotic systems have been considered [3], [4]. Most UAV's are underactuated systems, their coupled dynamics add a complexity to visual control problems. Many control laws were presented for aerial systems such as helicopters [5]–[8] for outdoor use as well as indoor operations [1], [9]. Visual servoing techniques could be classified into three main classes [10]: Position Based Visual Servo (PBVS or 3D), Image Based Visual Servo (IBVS or 2D) and the Homography Based Visual Servo (HBVS or $2\frac{1}{2}$D). 3D visual servoing needs a full reconstruction of the target pose with respect to the camera, it leads to a state estimation problem in the cartesian frame [11]–[13] and a classical state-space control design [5], [7], [8]. The main drawback of the PBVS methods is the need of a perfect knowledge of the target geometrical model [10], hence it is highly sensitive to camera calibration errors. The second class, known as 2D visual servoing, aims to control the dynamics of features directly in the image plane. Many extensions to the classical IBVS methods have been proposed for the control of non-linear dynamic systems, as the robust backstepping [1], [14] and optimal control techniques [15].

This paper is based on the homography method ($2\frac{1}{2}$D visual servoing) presented in [16], [17] that consists of combining 2D and 3D visual features, this method does not need an accurate model of the environment. More precisely, a homography matrix is estimated from the planar feature points extracted from two images (corresponding to the current and desired poses), and from this matrix, we estimate the relative position (translation vector and rotation matrix) of these two views. Many works have been in this line of thinking for robot manipulators [18]–[20] and wheeled mobile robots [21]. Homography-based strategies have succeeded to regulate the system's pose (position/orientation couple) to a constant position defined by a reference image. However using only one reference image results in some difficulties because the reference depth is an unobservable parameter [22]. In such cases, decoupling translation and rotation components could be useful. However, if depth information is needed another solution must be considered: let the system track a desired trajectory and design an adaptive update law to estimate depth information.

Due to new image technologies and advances in control, many researches have been interested lately to Trajectory Tracking. In [23], the authors proposed a visual tracking controller based on a linearized system of equations and Extended Kalman Filtering (EKF) techniques. Mahony and Hamel [24] considered a visual servo controller by tracking parallel linear visual features using nonlinear backstepping techniques.

This work could be viewed as an extension to the work done in [21] where the authors considered the kinematic equations of a mobile robot. In this paper, we consider a general mechanical dynamical model of a flying robot capable of quasi-stationary maneuvers. We will then derive a control law that forces the trajectory to track a prerecorded image sequences (desired trajectory). This desired trajectory could be taken from an operator-driven teach pendant step done preliminary. At each step the current image and the desired image will be compared to a reference image by homography matrices. To determine the full translation vector, we will estimate the reference depth information using the proposed adaptive control law. Unlike methods using EKF (Extended Kalman Filter), the Lyapunov-like analysis is based on the nonlinear dynamical model of the flying vehicle. The main contribution of this paper is a new method for visual servo

M. Metni is with Laboratoire Centrale des Ponts et Chaussées, Paris, France. (e-mail: metni@lcpc.fr)
M. Hamel is with I3S-CNRS, Nice-Sophia Antipolis, France.
F. Derkx is with Laboratoire Centrale des Ponts et Chaussées, Paris, France.

0-7803-9568-9/05/$20.00 ©2005 IEEE

controlling of a UAV in an unknown environment after an analysis of prerecorded image sequence. The method does not need any special predetermined landmarks, in addition the depth is estimated using an adaptive law. The major drawback is the lack of experimental results. In addition, the gravity cosine direction, which is an inertial measure, is computed from visual features under the condition that the gravity vector is orthogonal to the target. The outline of the paper is as follows: we present the mathematical model of a flying UAV in Section II, and the camera modelling is derived in Section III. The tracking control law for the complete dynamics and the adaptive update law are presented in Section IV. We provide simulations and results discussion in Section V.

II. UAV Dynamic Equations

In this section, we will derive mechanical equations for a general model of a UAV's in hover or quasi-stationary flights.

Let $\mathcal{F}^* = \{E_x, E_y, E_z\}$ denote a right-handed inertial or world frame such that E_z denotes the vertical direction downwards into the earth. Let $\xi = (x, y, z)$ denote the position of the center of mass of the object in the frame $\mathcal{F}^*$ relative to a fixed origin in $\mathcal{F}^*$. Let $\mathcal{F} = \{E_1^a, E_2^a, E_3^a\}$ be a (right-hand) body fixed frame. The orientation of the airframe is given by a rotation $R : \mathcal{F} \to \mathcal{F}^*$, where $R \in SO(3)$ is an orthogonal rotation matrix.

Let $V \in \mathcal{F}$ denote the linear velocity and $\Omega \in \mathcal{F}$ denote the angular velocity of the airframe both expressed in the body fixed frame. Let m denote the mass of the rigid object and let $\mathbf{I} \in \Re^{3\times3}$ be the constant inertia matrix around the center of mass (expressed in the body fixed frame $\mathcal{F}$). Using Newton formalism, it yields the following dynamic model for the motion of a rigid object:

$$\dot{\xi} = RV \tag{1}$$
$$m\dot{V} = -m\Omega \times V + F \tag{2}$$
$$\dot{R} = R\mathrm{sk}(\Omega), \tag{3}$$
$$\mathbf{I}\dot{\Omega} = -\Omega \times \mathbf{I}\Omega + \Gamma. \tag{4}$$

where F is the vector forces and Γ is the vector torques. The notation $\mathrm{sk}(\Omega)$ denotes the skew-symmetric matrix such that $\mathrm{sk}(\Omega)v = \Omega \times v$ for the vector cross-product $\times$ and any vector $v \in \Re^3$. The vector force F is defined as follows :

$$F = mgR^T e_3 - Te_3 \tag{5}$$

In the above equation, g is the acceleration due to gravity, and T represents the thrust magnitude, it is also the unique control input for the translational dynamics.

III. Camera Modelling and Homography Matrix

In this section we will present a brief discussion of the camera projection model and then introduce the homography relations.

A. Projection Models

Visual information is a projection from the 3D world to the 2D camera image surface. The pose of the camera determines a rigid body transformation from the current camera fixed frame $\mathcal{F}$ to the reference frame $\mathcal{F}^*$ and subsequently from the desired image frame $\mathcal{F}_d$ to $\mathcal{F}^*$. One has

$$P^* = RP + \xi \tag{6}$$
$$P^* = R_d P_d + \xi_d \tag{7}$$

as a relation between the coordinates of the same point in the current body fixed frame ($P \in \mathcal{F}$) and the desired body frame ($P_d \in \mathcal{F}_d$) with respect to the world frame ($P^* \in \mathcal{F}^*$). And where ξ and ξ_d are expressed in the reference frame $\mathcal{F}^*$.

Remark 3.1: There are 2 kinds of projection used in vision: the spherical and the flat projections. The spherical projection identifies the projection plane as the spherical surface and the image point p is given by $p = \frac{1}{|P|}(X, Y, Z)$. However, in the flat projection the point is projected on a plane with its image $p = \frac{1}{Z}(X, Y, Z)$. Indeed, since equality in projective geometry is an equality 'between directions', both points are on the same ray emanating from the origin and are thus not distinguished. In the following analysis, we will assume a calibrated camera but we do not distinguish between spherical or flat projections.

B. Planar Homography

Let p_i, p_{id} and p_i^* be the 3 images of the same point P' on the target when the camera is aligned respectively with the frames $\mathcal{F}$, $\mathcal{F}_d$ and $\mathcal{F}^*$.

Assuming we have a planar surface π containing a set of target points, the plane could be expressed as:

$$\pi = \left\{ P^* \in R^3 : {n^*}^T P^* - d^* = 0 \right\}$$

where d^* is the distance of the plane to the origin of F^*. n, n_d and n^* are the normal unit vectors to respectively the actual, the desired and the reference image planes. Let us define $t = -R^T\xi$ (resp. $t = -R^T\xi$). From equations 6 and 7 and since all target points lie in a single planar surface π, one has

$$p_i = \alpha \left(R^T + \frac{t{n^*}^T}{d^*} \right) p_i^*, \quad i = 1, \ldots, k \tag{8}$$
$$p_{id} = \alpha_d \left(R_d^T + \frac{t_d{n^*}^T}{d^*} \right) p_i^*, \quad i = 1, \ldots, k \tag{9}$$

The factors α and α_d are positive constants depending on the unknown parameter d^* which is the distance between the target and the desired plane. The projective mapping $H := \left(R^T + \frac{t{n^*}^T}{d^*} \right)$ (respectively $H_d := \left(R_d^T + \frac{t_d{n^*}^T}{d^*} \right)$) is called a homography matrix, it relates the images of points on a target plane when viewed from two different poses (defined by the coordinate systems $\mathcal{F}$ and $\mathcal{F}_d$ with respect to $\mathcal{F}^*$).Due to the inherent scale ambiguity in the term $\frac{1}{d^*}t$ in equation 8 the camera could only recover the ratio of the translation scaled by the inverse distance of the plane d^*. More details on the homography matrix could be found in [17]. The homography matrix contains the pose information (R, ξ) (resp. (R_d, ξ_d)) of the camera. To extract these information, many algorithms could be found in the literature (see for example [17], [25]–[27]). In general, the decomposition into (R, ξ) (resp. (R_d, ξ_d)) could be done up to two physically possible solutions. Depending on the application, some works presented a method for disambiguating, for example Shakernia *et al.* [7] studied the task of landing a UAV on a landing pad whose geometry is known *a priori*. This decomposition could consume a lot of resources (due to its complex calculations), hence we will propose a control strategy based on quantities extracted easily from the homography matrix. One quantity $r = \frac{d}{d^*}$ (resp. $r_d = \frac{d_d}{d^*}$)

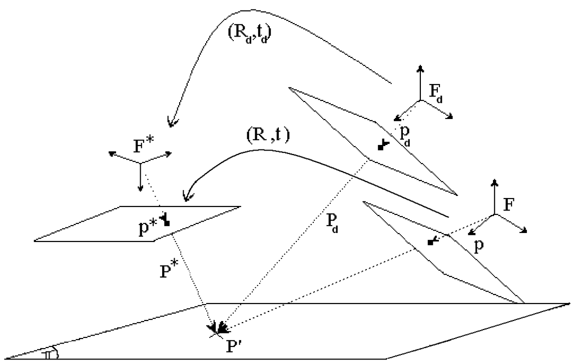

Fig. 1. Camera projection diagram showing the reference frame ($\mathcal{F}^*$), the current frame ($\mathcal{F}$) and the desired frame ($\mathcal{F}_d$)

could be calculated easily. the equation of the plane π could be written as $(n, P) = d$ for the usual inner product $(\cdot, \cdot)$. Thus $(n, R^T P^* + t) = d$ giving $(n, R^T P^*) = d - n^T t$. Therefore, $(Rn, P^*) = d - n^T t$ and it follows that:

$$n^* = Rn$$
$$d^* = d - n^T t$$

With changing the plane representation, we get the following relation:

$$r = 1 + \frac{n^T t}{d^*}$$

It can also be shown that

$$\det(H) = \det(R^T + \frac{t n^{*T}}{d^*}) = (1 + \frac{n^T t}{d^*}) = r$$

Similarly, we have $\det(H_d) = r_d$.

IV. Tracking Control Strategy

In the following analysis, it is assumed that the camera fixed frame coincides with the body frame. Let P' denote the observed point of reference of the planar target, and P^* be the representation of P' in the camera fixed frame at the reference position (Figure 1).

The control objective is to ensure that the coordinate frame $\mathcal{F}$ tracks the desired frame $\mathcal{F}_d$ (*i.e.*, the current image point p tracks the desired image point p_d). The tracking problem reduces to find a control input depending on the measured and the estimated states such that the errors

$$e_1(t) = (P - R^T P^*)$$
$$e_2(t) = (P_d - R_d^T P^*)$$

are asymptotically stable.

Note that the two error terms $e_1(t)$ and $e_2(t)$ are not defined in terms of visual information. Following [17], the camera can be controlled in the image space and in the Cartesian space at the same time. They propose the use of three independent visual features, such as the image coordinates of the target point associated with the ratio r delivered by determinant of the homography matrix. Consequently, let us consider the reference point P' lying in the reference plan π and define the scaled cartesian coordinates using visual information as follow:

$$P_r = \frac{{n^*}^T p^*}{n^T p} r p$$

Knowing that

$$\frac{||P||}{||P^*||} = \frac{{n^*}^T p^*}{n^T p} r,$$

it follows that we can reformulate the errors $e_1(t)$ and $e_d(t)$ in terms of the available visual information

$$\epsilon = \left(\frac{{n^*}^T p^*}{n^T p} r p - R^T p^* \right) \tag{10}$$

$$\epsilon_d = \left(\frac{{n^*}^T p^*}{n_d^T p_d} r_d p_d - R_d^T p^* \right) \tag{11}$$

From the above discussion and from eqn. 1 the dynamics of the errors are given by

$$\dot{\epsilon} = -\Omega \times \epsilon - \rho V \tag{12}$$

$$\dot{\epsilon}_d = -\Omega \times \epsilon_d - \rho V_d \tag{13}$$

where $\rho = \frac{1}{||P^*||}$ is an unknown parameter, it will be estimated by an adaptive update law using a double estimator [28]. The term V_d which is the velocity of the vehicle along the desired trajectory is an unknown term and could not be measured, however ϵ_d can be determined from the visual information. The desired trajectory will be used as a feed-forward in the control strategy. Note that to ensure the identification of ρ, the following assumptions must be satisfied:

1) The desired visual error ϵ_d is fourth order differentiable. This is due to the appearance of $\dot{\epsilon}_d$, $\ddot{\epsilon}_d$,$\epsilon_d^{(3)}$ and $\epsilon_d^{(4)}$ in the controller expression and the adaptive update law. For the sake of our analysis, the first three derivatives of ϵ_d must be known.
2) The desired visual error derivative $\dot{\epsilon_d}$ does not vanish for all $t > 0$.
3) Let $(\xi_d(t), R_d(t))$ be the desired trajectory which includes orientation information expressed in the inertial space. This trajectory (as well as the velocity V_d)could not be computed since the distance between the mobile

and the target is not measurable. Rather than working with the complete trajectory, one may choose R_d to be the identity rotation and then compute the visual variable ϵ_d.

4) To compute the gravity cosine direction from visual information, we assume that the plane π is perpendicular to the line of sight of the camera when aligned to the reference frame $\mathcal{F}^*$. In other terms, $n^* = e_3$.

Smooth desired visual error functions ϵ_d must be generated from the prerecorded image sequence. ϵ_d could be a smooth function of time and therefore derivatives would be easily extracted (*cf.* assumption 1)

We choose a given desired trajectory, and then look to a control law that achieves regulation of the error ($\delta_1 = \epsilon - R^T R_d \epsilon_d$) towards zero. Recalling assumption 3, the desired rotation will be chosen to be the identity rotation, then the error δ_1 could be written as

$$\delta_1 = \epsilon - R^T \epsilon_d$$

In addition to the basic tracking problem, it is desired that the control law estimates online the unknown value of ρ.

The dynamics of δ_1 is given by

$$\dot{\delta}_1 = -\Omega \times \delta_1 - \rho V - R^T \dot{\epsilon}_d$$

Following a standard trick in adaptive control when there is an unknown input gain, two dynamic variable estimates are introduced: $\hat{\rho}$ being the estimator of ρ and $\hat{b}$ the estimator of $b = \frac{1}{\rho}$. This procedure is used to avoid the division by $\hat{\rho}$ which could take a null value. We will choose a virtual input velocity V^v defined as

$$V^v = \hat{b}(\frac{k_1}{m}\delta_1 - R^T \dot{\epsilon}_d)$$

With this choice, one has

$$\dot{\delta}_1 = -\Omega \times \delta_1 - R^T \dot{\epsilon}_d - \rho\delta_2 - \rho V^v$$

where the error $\delta_2 = V - V^v$ is the difference between the velocity and its virtual input. Let $\tilde{\rho} = \rho - \hat{\rho}$ and $\tilde{b} = \frac{1}{\rho} - \hat{b}$ be the estimation errors. With the above choice of virtual control V^v, the time derivative of the error δ_1 becomes

$$\dot{\delta}_1 = -\Omega \times \delta_1 - \frac{k_1}{m}\delta_1 - \rho\delta_2 + \rho\tilde{b}\left(\frac{k_1}{m}\delta_1 - R^T \epsilon_d\right) \quad (14)$$

Define the following storage function

$$S_1 = \frac{1}{2}|\delta_1|^2 + \frac{1}{2\gamma_1}\rho\tilde{b}^2 \quad , \qquad \gamma_1 > 0$$

since the unknown constant ρ is positive, S_1 is positive definite in δ_1 and $\tilde{b}$. Taking the time derivative of S_1 one has

$$\dot{S}_1 = -\frac{k_1}{m}\delta_1^2 - \rho\delta_1^T\delta_2 + \rho\tilde{b}\delta_1^T\left(\frac{k_1}{m}\delta_1 - R^T\epsilon_d\right) - \frac{\rho}{\gamma_1}\dot{\hat{b}}\tilde{b}$$

Cancelling the terms containing the unknown error $\tilde{b}$, we choose the following dynamics for the estimator $\hat{b}$

$$\dot{\hat{b}} = \gamma_1\delta_1\left(\frac{k_1}{m}\delta_1 - R^T\dot{\epsilon}_d\right) \quad (15)$$

with the above choice, one has

$$\dot{S}_1 = -\frac{k_1}{m}\delta_1^2 - \rho\delta_1^T\delta_2$$

Deriving δ_2 and recalling Eq.15

$$\begin{aligned}\dot{\delta}_2 =& \dot{V} - \dot{V}^v \\ =& -\Omega \times \delta_2 - \frac{k_2}{m}\delta_2 + \delta_3 + \frac{k_1}{m}\hat{b}\tilde{\rho}V + \hat{\rho}\delta_1 \quad (16)\end{aligned}$$

where the error $\delta_3 = F - F^v$ is the difference between the input and the virtual input F_v which is given by

$$\begin{aligned}\frac{F^v}{k_1} =& \dot{\hat{b}}(\frac{k_1}{m}\delta_1 - R^T\dot{\epsilon}_d) - \frac{k_1}{m}\hat{b}\hat{\rho}V - \frac{k_1}{m}\hat{b}R^T\dot{\epsilon}_d \\ & - \hat{b}R^T\ddot{\epsilon}_d + \hat{\rho}\delta_1 - \frac{k_2}{m}\delta_2 \quad (17)\end{aligned}$$

At this stage we define a second storage function S_2

$$S_2 = \frac{1}{2}|\delta_1|^2 + \frac{1}{2}|\delta_2|^2 + \frac{1}{2\gamma_1}\rho\tilde{b}^2$$

and its time derivative given by

$$\dot{S}_2 = -\frac{k_1}{m}\delta_1^2 - \frac{k_1}{m}\delta_2^2 - \tilde{\rho}\delta_1^T\delta_2 + \delta_2^T\delta_3 + \frac{k_1}{m}\hat{b}\tilde{\rho}\delta_2 V$$

In terms of the error variables, the δ_2 derivative may be written as

$$\dot{\delta}_2 = -\Omega \times \delta_2 + \hat{\rho}\delta_1 - \frac{k_2}{m}\delta_2 + \delta_3 + \frac{k_1}{m}b\tilde{\rho}V \quad (18)$$

To continue with the backstepping procedure, we derive Eq.17 to get the dynamics of δ_3 which is quite complex due to its dependance on many parameters, we will present the time derivative of δ_3 in a simple form

$$\dot{\delta}_3 = -\Omega \times \delta_3 + \Omega \times F + \dot{F} - Y - X\tilde{\rho} - A\dot{\hat{\rho}} \quad (19)$$

where X is the part related to the unknown variable $\tilde{\rho}$, Y gathers almost all known or measurable terms and A is the part multiplying $\dot{\hat{\rho}}$. X, Y and A are functions of all known parameters (see the appendix for complete equations). Applying two operations (derivation and cross multiplying by Ω) to Eq.5 and then adding

$$\Omega \times F + \dot{F} = -\dot{T}e_3 + T\mathrm{sk}(e_3)\Omega$$

Choosing the virtual input to be:

$$[-\dot{T}e_3 + T\mathrm{sk}(e_3)\Omega]^v = A\dot{\hat{\rho}} + Y - \delta_2 - k_3\delta_3 \quad (20)$$

Knowing that $e_3 \in \ker[\mathrm{sk}(e_3)]$, the two terms of the above equation are independent, we will separate them by multiplying Eq.20 by e_3

$$\dot{T} = -e_3^T\left(A\dot{\hat{\rho}} + Y - \delta_2 - k_3\delta_3\right) \quad (21)$$

and then multiply Eq.20 by $\pi_{e_3} = \mathrm{I} - e_3 e_3^T$

$$[T\mathrm{sk}(e_3)\Omega]^v = \pi_{e_3}\left(A\dot{\hat{\rho}} + Y - \delta_2 - k_3\delta_3\right) \quad (22)$$

The storage function associated with this stage of backstepping is

$$S_3 = \frac{1}{2}|\delta_1|^2 + \frac{1}{2}|\delta_2|^2 + \frac{1}{2}|\delta_3|^2 + \frac{\rho}{2\gamma_1}\tilde{b}^2 + \frac{1}{2\gamma_2}\tilde{\rho}^2 \quad (23)$$

Taking the derivative of S_2 yields

$$\begin{aligned}\dot{S}_3 = &- \frac{k_1}{m}|\delta_1|^2 - \frac{k_2}{m}|\delta_2|^2 - \frac{k_3}{m}|\delta_3|^2 + \delta_3^T\delta_4 \\ &+ \frac{k_1}{m}\hat{b}\tilde{\rho}\delta_2^T V - \tilde{\rho}(\delta_3^T X + \delta_1^T\delta_2) - \frac{1}{\gamma_2}\tilde{\rho}\dot{\hat{\rho}}\end{aligned} \tag{24}$$

where δ_4 is the input error

$$\delta_4 = T\mathrm{sk}(e_3)\Omega - [T\mathrm{sk}(e_3)\Omega]^v$$

To cancel the contribution of the parametric error $\tilde{\rho}$ in Eq.24, we choose the dynamics of $\hat{\rho}$ as:

$$\dot{\hat{\rho}} = \gamma_2\left[\frac{k_1}{m}\hat{b}\delta_2^T V - \delta_1^T\delta_2 - \delta_3^T X\right] \tag{25}$$

The last step of this procedure is to compute the torque control. For the sake of simplicity, we will use the high gain control. Consider the derivative of δ_4

$$\dot{\delta_4} = \dot{T}\mathrm{sk}(e_3)(\Omega - \Omega^v) + T\mathrm{sk}(e_3)(\dot{\Omega} - \dot{\Omega}^v)$$

We will choose a control law for $\dot{\Omega}$ with a gain high enough to neglect the effect of $\dot{\Omega}^v$. Let

$$\dot{\Omega} = \left(-k_4 + \frac{\dot{T}}{T}\right)(\Omega - \Omega^v) \tag{26}$$

with $\Omega^v = [\Omega_1^v\ \Omega_2^v\ \Omega_3^v]^T$. The first two terms Ω_1^v and Ω_2^v could be computed from Eq. 22, and Ω_3 will be extracted from Eq.28. Recalling Eq.4, the torque input Γ is introduced via the derivative $\dot{\Omega}$

$$\Gamma = \mathbf{I}\dot{\Omega} + \Omega \times \mathbf{I}\Omega \tag{27}$$

The dynamical structure of this kind of flying vehicle and the appropriate backstepping control strategy requires only the control inputs $\dot{T}$, Γ to achieve the desired trajectory tracking. This leaves the input Ω_3 free to stabilize the yaw speed from the following equation:

$$\dot{\Omega}_3 = -k_5\Omega_3\ , \qquad k_5 > 0 \tag{28}$$

Then the proposed control algorithm will achieve the monotonic decrease of the following Lyapunov function

$$\mathcal{L} = \frac{1}{2}|\delta_1|^2 + \frac{1}{2}|\delta_2|^2 + \frac{1}{2}|\delta_3|^2 + \frac{1}{2}|\delta_4|^2 + \frac{\rho}{2\gamma_1}\tilde{b}^2 + \frac{1}{2\gamma_2}\tilde{\rho}^2 + \frac{1}{2}\Omega_3^2$$

and its time derivative given by

$$\begin{aligned}\dot{\mathcal{L}} = &- \frac{k_1}{m}|\delta_1|^2 - \frac{k_2}{m}|\delta_2|^2 - \frac{k_3}{m}|\delta_3|^2 - \frac{k_4}{m}|\delta_4|^2 \\ &+ \delta_3^T\delta_4 - k_5\Omega_3^2\end{aligned} \tag{29}$$

the above equation is negative definite if the following conditions are satisfied

$$k_1 > 0 \tag{30}$$

$$k_2 > 0 \tag{31}$$

$$k_3 > 0 \tag{32}$$

$$k_4 > \frac{m^2}{2k_3} \tag{33}$$

Theorem 4.1: Consider the dynamics of the flying vehicle. Let the control $\dot{T}$ and Γ be given by Eqs. 21 and 27. In addition let all the conditions given by Eqs. 30 to 33 be satisfied. Therefore, the proposed control algorithm ensures the asymptotic convergence of the error δ_1 and the exponential stability of Ω_3. In addition, the control law ensures the convergence of the parameters to their true values:

$$\tilde{b} \to 0\ , \quad \tilde{\rho} \to 0$$

Proof: Applying Lyapunov argument in Eq.29, one can conclude that the errors δ_1, δ_2, δ_3 and δ_4 converge asymptotically to zero. From Eq. 28, one can ensure the exponential stability of Ω_3.

To prove the convergence of the estimator parameter errors $\tilde{b}$ and $\tilde{\rho}$, we appeal to LaSalles principle. The invariant set is contained in the set defined by the conditions $\delta_i \equiv 0$ $(i = 1, 2, 3, 4)$. Recalling Eqs.15 and 25, it follows that $\dot{\hat{b}} = 0$ and $\dot{\hat{\rho}} = 0$ on the invariant set. Taking the expressions of the derivatives of the errors δ_1 and δ_2, it follows that

$$\rho\tilde{b}\dot{\epsilon_d} = 0 \tag{34}$$

$$\tilde{\rho}\hat{b}^2\dot{\epsilon_d} = 0 \tag{35}$$

From equation 34 and knowing that ρ is a constant, and under the assumption that $\dot{\epsilon_d} \neq 0$ on the invariant set, we ensure the convergence of $\tilde{b}$ to zero. In this way $\hat{b}$ will converge to a constant b. The second equation (Eq.35) ensures the convergence of $\tilde{\rho}$ to zero ($\hat{b}$ is a constant). Consequently, this ensures the asymptotical convergence of $\hat{b}$ and $\hat{\rho}$ to their true values. ■

V. Simulation Results

In this section, we present a simulation example in order to evaluate the effectiveness of the proposed control and estimation laws. The experiment considers a desired trajectory defined in the image plane as a circle centered at $(0, 0, 5)$ and of radius 1. The points p_{id} and the corresponding parameters were computed off line before the start of the tracking mission. The reference image is composed from five points: four on the vertices of a planar square and one on its center. The available information are the pixel coordinates of the five points observed by the camera.

The parameters used for the dynamic model are $m = 1.5$, $\mathbf{I} = \mathrm{diag}[0.4, 0.4, 0.6]$ and $g = 10$. Initially, the robot is assumed to hover at a position $(5, 4, 12)$. It is assumed that the plane of the reference image is parallel with the target plane at a distance $b = 3$ (*i. e.* $\rho = 1/3$ and the unit vector normal to the target plane is equal to the direction of the gravity $n^* = e_3$ as mentioned in assumption 4). For the adaptive update law, the initial guesses of $\hat{\rho}_0$ and $\hat{b}_0$ are as follows:

$$\hat{\rho}_0 = 0.25$$

$$\hat{b}_0 = \frac{1}{\hat{\rho}_0}$$

The control design used the following gains,

$$k_1 = 2.5,\, k_2 = 1.2,$$
$$k_3 = 2.0,\, k_4 = 2.5,$$
$$\gamma_1 = 1/400,\, \gamma_2 = 1/80$$

The gains γ_1 and γ_2 of the adaptive law must be chosen very carefully. Some tuning must be performed to choose the control gains then adjust the adaptive gains.

In figures 2, 3 and 4 are the results of the simulation described above. From figure 2, it is clear that the position of the vehicle is tracking smoothly a circle contained in the desired plan $z = 5$. Figure 3 described the trajectory of the image in the image plane , it shows more explicitly the convergence to the desired trajectory defined by complete circles of radius $r = 1$. Figure 4 shows us the convergence of the estimations of ρ and b to the exact values ($\rho = \frac{1}{3}$ and $b = 3$).

To show the robustness of our estimators, a white noise was added to the image acquisition process to simulate the input noise as well as the disturbances encountered in an unknown environment (low and high frequencies). Figure 5 shows the results of the two estimations $\hat{\rho}$ and $\hat{b}$ along the mission. It may be seen that the added noise did not degrade the performance of the estimators. The two estimations converge, with some fluctuations, to their true values.

In these simulations, the desired trajectory was totally known and incorporated in the algorithm as a circle equation. We did not take into account the point matching problem and all its subsequent difficulties as feature points loss and reselecting new points. We thus assume here that all features points must always stay in the camera's field of view.

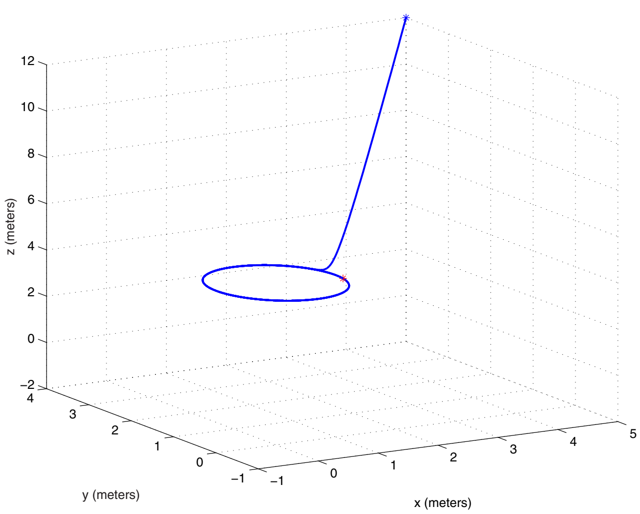

Fig. 2. Evolution of the position of the mobile

VI. Conclusion

In this paper, we have proposed a control law to force an unmanned aerial vehicle (UAV) to track a desired trajectory defined by a series of prerecorded images. Euclidean homographies were extracted using three views: a current image, the corresponding desired image and a unique reference image. Extracting the pose parameters from the homographies will leave us with unknown parameters depending on the depth from the target to the reference image. Thus, an adaptive update law for the estimation of this unknown constant parameter was also presented. Simulations are provided to prove the convergence of the estimator as well as the controller.

This work is part of a research direction in the autonomous UAV's flights where a big work still to be done. In addition, the linear and rotational velocities of the vehicle must be accurately known, in this field the authors are also working on nonlinear state observers.

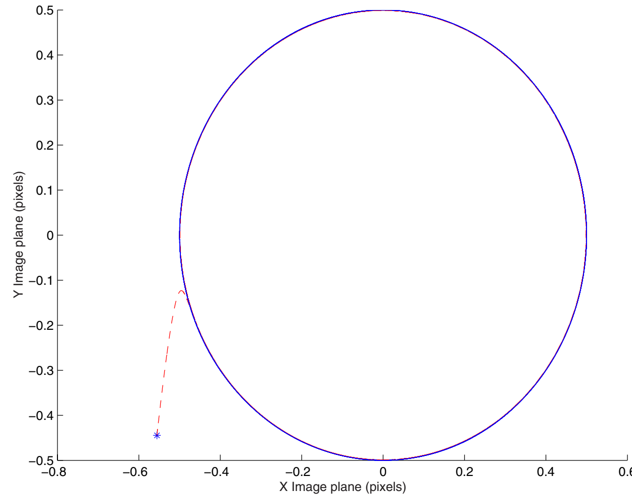

Fig. 3. Evolution of the point on the image plane

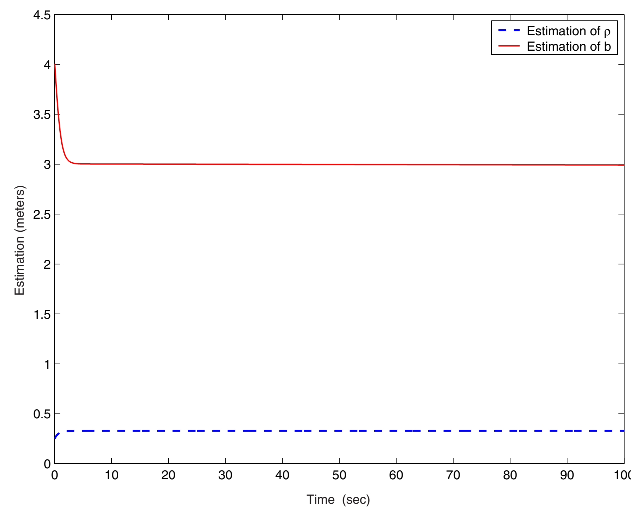

Fig. 4. Evolution of 2 estimations: $\hat{\rho}$ (dashed line) and $\hat{b}$ (solid line)

References

[1] T. Hamel and R. Mahony, "Visual servoing of under-actuated dynamic rigid body system : An image based approach," *IEEE Transactions on Robotics and Automation*, vol. 18, no. 2, pp. 187–198, 2002.

[2] F. Shell and E. Dickmanns, "Autonomous landing of airplanes by dynamic machine vision," *Machine Vision and Applications*, vol. 7, pp. 127–134, 1994.

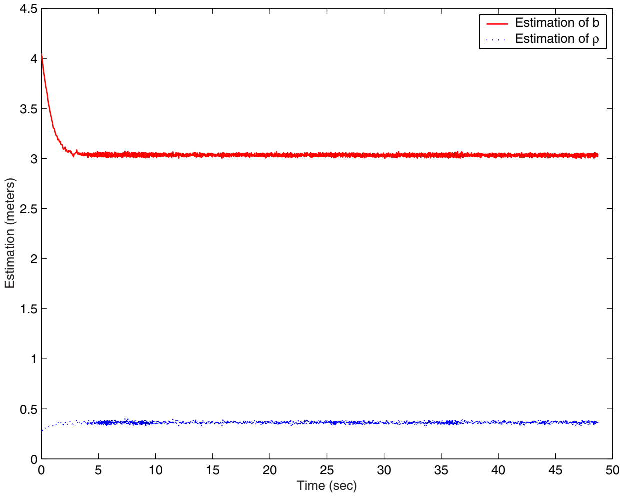

Fig. 5. Evolution of 2 estimations: $\hat{\rho}$ (dashed line) and $\hat{b}$ (solid line) with added white noise to image acquisition

[3] C. Conticelli, B. Allotta, and P. Khosla, "Image-based visual servoing of nonholonomic mobile robots," in *Proc. Conference on Decision and Control*, Phoenix, Arizona, USA, 1999.

[4] Y. Ma, J. Koseck'a, and S. Sastry, "Vision guided navigation for a nonholonomic mobile robot," *IEEE Transactions on robotics and automation*, vol. 15, no. 3, pp. 521–536, 1999.

[5] O. Amidi, T. Kanade, and R. Miller, "Vision-based autonomous helicopter research at carnegie mellon robotics institute (1991-1998)," in *Robust Vision for Vision-Based Control of Motion*, M. Vincze and G. Hager, Eds. New York, USA: IEEE press and SPIE Optical Engineering press, 1999, ch. 15, pp. 221–232.

[6] M. Dahlen, E. Frazzoli, and E. Feron, "Trajectory tracking control design for autonomous helicopters using a backstepping algorithm," in *American Control Conference ACC*, Chicago, Illinois, USA, 2000, pp. 4102–4107.

[7] O. Shakernia, Y. Ma, T. Koo, and S. Sastry, "Landing an unmanned air vehicle: Vision based motion estimation and nonlinear control," *Asian Journal of Control*, vol. 1, no. 3, pp. 128–146, 1999.

[8] S. Saripalli, J. Montgomery, and G. Sakhatme, "Vision based autonomous landing of an unmanned aerial vehicle," in *Proc. International Conference of Robotics and Automation, ICRA 2002*, Washington DC, Virginia, USA, 2002.

[9] E. Altug, J. Ostrowski, and R. Mahony, "Control of quadrotor helicopter using visual feedback," in *Proc. IEEE International Conference on Robotics and Automation, ICRA 2002*, Washington DC, Virginia, USA, 2002.

[10] S. Hutchinson, D. Hager, and P. Cake, "A tutorial on visual servo control," *IEEE Transactions on Robotics and Automation*, vol. 12, no. 5, October 1996.

[11] S. Soatto and P. Perona, "Structure-independant visual motion control on the essential manifold," in *Proc. IFAC Symposium on Robot Control, SYROCO 94*, Capri, Italy, 1994, pp. 869–876.

[12] Y. Ma, J. Koseck'a, and S. Sastry, "Linear differential algorithm for motion recovery: A geometric approach," *International Journal of Computer Vision*, pp. 92–99, 2002.

[13] T. Fitzgibbons and E. Nebot, "Application of vision in simultaneous localization and mapping," *Intelligent Autonomous Systems*, vol. 36, no. 1, pp. 71–89, 2000.

[14] E. Zergeloglu, D. Dawson, M. de Queiroz, and S. Nagarkatti, "Robust visual-servo control of robot manipulators in the presence of uncertainty," in *Proc. 38th COnference on Decision and Control*, Phoenix, Arizona, USA, 1999.

[15] H. Zhang and J. Ostrowski, "Visual servoing with dynamics: Control of an unmanned blimp," *IEEE Transactions on Robotics and Automation*, vol. 18, no. 2, April 2002.

[16] E. Malis and F. Chaumette, "Theoretical improvements in the stability analysis of a new class of model-free visual servoing methods," *IEEE Transactions on Robotics and Automation*, vol. 18, no. 2, pp. 176–186, April 2002.

[17] E. Malis, F. Chaumette, and S. Bodet, "$2\frac{1}{2}$d visual servoing," *IEEE Transactions on Robotics and Automation*, vol. 15, no. 2, pp. 238–250, April 1999.

[18] P. Corke and S. Hutchinson, "A new hybrid image-based visual servo control scheme," in *Proc. IEEE Conference on Decision and Control*, 2000, pp. 2521–2527.

[19] K. Deguchi, "Optimal motion control for image-based visual servoing by decoupling translation and rotation," in *Proc. of the Intl. Conf. on Intelligent Robots and Systems*, 1998, p. October.

[20] Y. Fang, A. Bahel, W. Dixon, and D. Dawson, "Adaptive 2.5d visual servoing of kinematically redundant robot manipulators," in *Conference on Decision and Control*, Las Vegas, NV, USA, 2002, pp. 2860–2865.

[21] J. Chen, W. Dixon, D. Dawson, and M. McIntire, "Homography-based visual servo tracking control of a wheeled mobile robot," *IEEE Transactions on Robotics*, To appear.

[22] N. Metni, T. Hamel, and F. Derkx, "A uav for bridges inspection: Visual servoing control law with orientation limits," in *5th Symposium on Intelligent Autonomous Vehicles (IAV 04)*, Lisboa, Portugal, July 2004.

[23] B. Kim *et al.*, "Localization of a mobile robot using images of a moving target," in *Proc. IEEE International Conference on Robotics and Automation*, 2001, pp. 253–258.

[24] R. Mahony and T. Hamel, "Visual servoing using linear features for under-actuated rigid body dynamics," *Proc. IEEE/RJS International Conference on Intelligent Robots and Systems*, pp. 1153–1158, 2001.

[25] J. Weng, T. Huang, and N. Ahuja, *Motion and Structure from Image Sequences*. Springer-Verlag, 1993.

[26] Z. Zhang and A. Hanson, "Scaled euclidean 3d reconstruction based on externally uncalibrated cameras," in *Proc. IEEE Symposium on Computer Vision*, Coral Glabes, Floride, USA, 1995.

[27] J. Weng, T. Huang, and N. Ahuja, "Motion and structure from line correspondences: Closed-form solution, uniqueness, and optimization," *Proc. IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 14, no. 3, March 1992.

[28] M. Kristc, I. Kanellakopoulos, and P. Kokotovic, *Nonlinear and Adaptive Control Design*. Rhode Island, USA: American Mathematical Society, 1995.

APPENDIX

This appendix gives the complete equation of the derivative of δ_3. Recalling Eq.19

$$\dot{\delta}_3 = -\Omega \times \delta_3 + \Omega \times F + \dot{F} - Y - X\tilde{\rho} - A\dot{\hat{\rho}}$$

The three expressions of A, X and Y are given by

$$\begin{aligned}
A &= -\frac{k_1}{m}\hat{b}V + \delta_1 \\
X &= \left(\ddot{\hat{b}}\right)^u \left(\frac{k_1}{m}\delta_1 - R^T\dot{\epsilon}_d\right) - \left(\frac{k_1}{m}\dot{\hat{b}} - \hat{\rho} - \frac{k_1 k_2}{m}\hat{b}\right) V \\
Y &= \left(\ddot{\hat{b}}\right)^k \left(\frac{k_1}{m}\delta_1 - R^T\dot{\epsilon}_d\right) - \hat{b}R^T\dot{\epsilon}_d - \frac{k_1}{m}\dot{\hat{b}}\hat{\rho}V \\
&\quad + \left(\hat{\rho}V + R^T\dot{\epsilon}_d\right)\left(-\frac{k_1}{m}\dot{\hat{b}} + (\frac{k_1}{m})^2\hat{b}^2\hat{\rho} - \hat{\rho}\right) \\
&\quad - \left(\frac{k_1}{m} + 1\right)\left(\dot{\hat{b}}R^T\dot{\epsilon}_d + \hat{b}R^T\ddot{\epsilon}_d\right) \\
&\quad - \frac{k_1}{m}\hat{b}\hat{\rho}\left[\dot{\hat{b}}\left(\frac{k_1}{m}\delta_1 - R^T\dot{\epsilon}_d\right) - \hat{b}R^T\ddot{\epsilon}_d\right] \\
&\quad - \left(\frac{k_2}{m} + \frac{k_1}{m}\hat{b}\hat{\rho}\right)\left(\hat{\rho}\delta_1 - \frac{k_2}{m}\delta_2 + \delta_3\right)
\end{aligned}$$

The notation $\left(\ddot{\hat{b}}\right)^k$ and $\left(\ddot{\hat{b}}\right)^u$ denote respectively the known (or measurable) and unknown parts of the expression of $\ddot{\hat{b}}$. In other terms, $\ddot{\hat{b}}$ could be written as

$$\ddot{\hat{b}} = \left(\ddot{\hat{b}}\right)^k + \tilde{\rho}\left(\ddot{\hat{b}}\right)^u$$