HAL
open science

FULL MODELLING AND SLIDING MODE CONTROL FOR A QUADROTOR UAV IN VISUAL SERVOING TASK

Choukri Bensalah, Nacer K M'Sirdi, Aziz Naamane

To cite this version:

Choukri Bensalah, Nacer K M'Sirdi, Aziz Naamane. FULL MODELLING AND SLIDING MODE CONTROL FOR A QUADROTOR UAV IN VISUAL SERVOING TASK. IMAACA2019, Feb 2019, Lisbon, Portugal. hal-02471653

HAL Id: hal-02471653

https://hal.science/hal-02471653v1

Submitted on 8 Feb 2020

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.

FULL MODELLING AND SLIDING MODE CONTROL FOR A QUADROTOR UAV IN VISUAL SERVOING TASK

Choukri Bensalah[2], Nacer K. M'Sirdi[1], Aziz Naamane[1]

[1]Aix Marseille University, Université de Toulon, CNRS, LIS UMR 7020, SASV, Marseille, France

[2]Laboratoire d'Automatique de Tlemcen (LAT), University of Tlemcen, Algeria.

[1]nacer.msirdi@lis-lab.fr, [1]aziz.naamane@lis-lab.fr , [2]choukri.bensalah@lis-lab.fr

ABSTRACT

In this paper the Control of an UAV is designed by an Image Based Visual Servoing (IBVS). The trajectories are generated by the IBVS to track a target. The dynamic model describing the Quadrotor behaviour is established taking into account all internal and external aerodynamic forces and moments, for simulation. Furthermore, actuators and sensors dynamics are also considered. To overcome all the nonlinearities, as well as the strong coupling in 3D position and Euler angles of Quadrotor system, a robust Sliding Mode Control (SMC) is designed. Unlike several literature works in this topic, the SMC control uses only an approximated model of the Quadrotor. In other words, SMC does not use the dynamic inversion of Quadrotor model. The Quadrotor is endowed a virtual camera to perform visual tracking, in order to evaluate the robustness of our controller.

Keywords : Full Quadrotor modelling, SMC Control, nonlinear control and compensation, IBVS, Virtual Camera.

1. INTRODUCTION

Unmanned Aerial Vehicles (UAV) are popular due to the multiple application possibilities (rescue, surveillance, inspection, mapping, bridges, buildings supervision, and cinema). Their ability to take off and land vertically, to perform stationary flight as well as their manoeuvrability and controllability give them a key position as mobile robots Guerrero and Lozano [2012]Rabhi et al. [2011]Lozano [2013]. The four rotors helicopter (see figure 1) exhibits a nonlinear behaviour which is subject to aerodynamic forces, moments and unknown disturbances Mistler et al. [2001]Mistler et al. [2002]. It is multi-variable, non linearly coupled and has inherent uncertainties in both high and low frequencies Mederreg et al. [2004]. This work was supported by SASV of LiS from AMU

The required application objective is its capability to follow desired trajectories and autonomous motions. The system must have some desired features like robustness to uncertainties, perturbations and parameters variations Mokhtari et al. [2006].

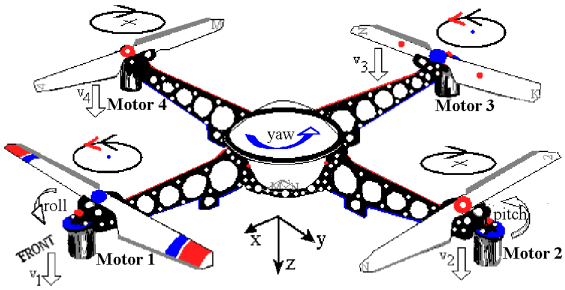

Figure 1: Mechanical structure of the Quadrotor.

During the past two decades, the Quadrotor is considered as suitable platform to evaluate and compare the performances of designed controllers. In his thesis Bouabdallah [2007], the author has developed several existing classical controllers; Lyapunov based control, PID, LQR, Backstepping, SMC, Adaptive optimal control and others obtained by combining the approaches. To overcome the under-actuated problem the adopted control consists of acting directly on the three Euler angles and the altitude using the four inputs. Two non-actuated variables are controlled by Virtual Inputs related to both pitch and roll angles. In Mokhtari et al. [2006], the authors use a feedback linearization leading to a linear extended system, based on the approach of Mistler et al. [2001]. Other strategies have been proposed using neural network Das et al. [2008], fuzzy logic approach Cosmin and Macnab [2006], dynamic inversion mechanism Das et al. [2009] or H_∞ robust control Raffo et al. [2011].

The most of the literature works, use for the control the same dynamic equations as the simulated model. However, it is not possible to know exactly this dynamic. Furthermore, to prove the robustness of controllers, the authors introduce disturbances during the simulation or, they assume uncertainties in some (but not all) parameters, neither errors on all the measured outputs Becker et al. [2012].

In B. Wang [2017], Islam et al. [2015], Emelyanov [2007] for example, the modelling error and disturbances uncertainties are considered by assuming knowledge of theirs bounds and model structure.

Sliding mode control approach, which is purely nonlinear, is dedicated to a wide spectrum of systems, especially for Variable Structure Systems Emelyanov [2007]. The outstanding feature of this controller is its robustness with regard to model error, parameters' uncertainties, external disturbances and noisy measurements, the reasons of which the SMC is adopted in our work. In addition to that, SMC is selected to be able to deal with arisen effects during the motors control loop M'Sirdi and Nadjar-Gauthier [2002].
From the most popular methods, the rotors speed is used as input reference for motor control by a linear loop Chan and Woo [2015]. In our case, a new contribution consists of using directly the thrust force as input reference for motor control. This means that the forces are controlled directly, which compensate errors due to non linearity of actuation, as what is done for pneumatic or hydraulic actuators Manamanni et al. [2001], M'Sirdi et al. [1997].
On the other side, the performance of SMC could be affected due to the succession of two sliding mode controllers in cascade (inner and outer control loops). This problem is avoided by considering two different convergence rate. To evaluate the effectiveness of the designed SMC, the Quadrotor performs trajectories tracking generated by visual servoing controller. This last controller provides a potential technique to fulfill motions of UAV thanks to visual measurement extracted from a captured image of an inboard camera. In the literature of UAV controlled by visual servoing task, there are two operating principles Hamel and Mahony [2006], Azrad et al. [2010];
- position-based visual servoing (PBVS) and
- image-based visual servoing (IBVS).
The implementation of IBVS is easier than PBVS since it is not necessary to know and reconstruct the 3D model of the target Chaumette and Hutchinson [2006].
This is one of the motivations to select IBVS method through this paper. Furthermore, the dynamic and the projection principle of IBVS controller are based on the virtual camera concept. Usually, the velocity trajectories generated by IBVS are compared with the velocities, mostly provided from Inertial Measurement Unit (IMU) sensor. The formulation background used is the Task Function approach Samson and Espiau [1990]. However, the IMU sensor exhibits bias and noise in the measurements which can cause a non-zero steady state error in image information Xie et al. [2017]. To overcome the problem, it is possible to obtain the velocity measurement only from the optical flow using the image information Mahony et al. [2017], Herisse et al. [2012] or to combine optical flow and IMU measurements Grabe et al. [2013].
This paper is organized as follows. We start with a full modelling of the Quadrotor in which all effects internal and external, forces and moments are considered. To simulate IMU measurements, we consider a dynamics for sensors taking into account a measurement bias and noise. In Section III, we explain different steps to design SMC. In Section IV, IBVS controller based on a virtual camera is detailed. SMC under IBVS controller for tracking task is simulated in Section V. Finally, Section VI reports some conclusions of the paper.

2. QUADROTOR UAV MODELLING

2.1. System description

The four rotors helicopter, shown in Fig. 1, is propelled by the four forces F_i $(i \in \{1,2,3,4\})$. The UAV is moved by varying the rotor speeds. The impair rotors (1, 3) turn in the same direction, which is in opposite directions of the pair ones (2, 4). This eliminates the anti-torque.
A variation of the rotor speeds altogether with the same quantity creates the lift forces which will affect the altitude z enabling vertical take-off or landing. The velocity speeding up or slowing down the diagonal motors creates the moment which produce a yaw or pitch motion. Yaw angle y is obtained by speeding up or slowing down the clockwise motors regard to the others. Pitch angle j axis allows the Quadrotor to move towards the longitudinal direction x. Roll angle f allows the Quadrotor to move towards the lateral direction y.

2.2. Actuators model

The thrust forces F_i are generated by 4 DC motors M_i $(i \in \{1,2,3,4\})$ as depicted in Fig. 1. These forces are assumed to be proportional to the square of the angular motors speeds, denoted with ω_i and is given by:

$$F_i = b\omega_i^2 \tag{1}$$

where $b = \frac{1}{2}\rho\Lambda C_T r^2$ with ρ is the air density, r and Λ are the radius and the section of the propeller, respectively. The term C_T is the aerodynamic thrust coefficient. $\omega_{1,2,3,4}$ are the angular speeds of the four rotors. The aerodynamic drag torques δ_i produced at each actuator are opposed to the motor torque and proportional to the propeller angular speed.

$$\delta_i = \frac{1}{2}\rho\Lambda C_D r^2 \omega_i^2 = d\omega_i^2 \tag{2}$$

where C_D is the aerodynamic drag coefficient.
Therefore, these forces create different torques around the pitch ϕ, roll θ and yaw ψ axis that are respectively given as follows (Guerrero and Lozano [2012]; Austin [2010]):

$$\Gamma_\theta = u_2 = l\,(F_3 - F_1) \tag{3}$$

$$\Gamma_\phi = u_3 = l\,(F_4 - F_2) \tag{4}$$

$$\Gamma_\psi = u_4 = d\,(F_1 - F_2 + F_3 - F_4) \tag{5}$$

where d is a positive coefficient defined in Eq.2 and l is the distance between the generated force position and the gravity center of the Quadrotor.
Recall that the sum of the forces control the vertical motion z. So, the Quadrotor is controlled by varying the speed of these four motors. The control inputs of the

Quadrotor dynamics are thus defined as follows, with $u = (u_1, u_2, u_3, u_4)^T$:

$$\begin{pmatrix} u_1 \\ u_2 \\ u_3 \\ u_4 \end{pmatrix} = \begin{pmatrix} 1 & 1 & 1 & 1 \\ 0 & -l & 0 & l \\ -l & 0 & l & 0 \\ d & -d & d & -d \end{pmatrix} \begin{pmatrix} F_1 \\ F_2 \\ F_3 \\ F_4 \end{pmatrix} \quad (6)$$

u_1 denotes the first input force of the Quadrotor body in the z-axis. The inputs u_2 and u_3 represent the roll and the pitch input torques, respectively. The input u_4 represents the yawing control torque. These forces are provided through four Brushless DC motors which are characterized by a high torque and little friction (Austin [2010]). The dynamic model equation of all motors are reported in the next subsection.

2.3. Kinematics and Dynamics

The Quadrotor is described through the body-frame $\mathbf{R}_B(O, x_b, y_b, z_b)$ and earth-frame $\mathbf{R}_E(o, x_e, y_e, z_e)$ as shown in Fig. 1. Let us note $\xi = (x, y, z)^T$ the absolute Cartesian position of the Quadrotor Center of Gravity (CoG) relative to its fixed earth-frame $\mathbf{R}_E$ and, the Euler angles $\eta = (\phi, \theta, \psi)^T$ give its attitude relative to $\mathbf{R}_E$. The rotation matrix $\mathscr{R} : \mathbf{R}_E \to \mathbf{R}_B$ depends on the Euler angles (ϕ, θ, ψ) Mistler et al. [2001]. It is defined as follows:

$$\mathscr{R}(\phi, \theta, \psi) = \begin{pmatrix} c\psi c\theta & s\phi s\theta c\psi - s\psi c\theta & c\phi s\theta c\psi + s\psi s\phi \\ s\psi c\theta & s\phi s\theta s\psi + c\psi c\theta & c\phi s\theta s\psi - s\phi c\psi \\ -s\theta & s\phi c\theta & c\phi c\theta \end{pmatrix} \quad (7)$$

where $c = \cos(.)$ and $s = \sin(.)$ and under the stability limit constraints on the pitch $\phi \in \left[-\frac{\pi}{2}, \frac{\pi}{2}\right]$, the roll $\theta \in \left[-\frac{\pi}{2}, \frac{\pi}{2}\right]$, pitching and the yaw $\psi \in [-\pi, \pi]$ motions. The linear velocity of the UAV in the earth-frame $\mathbf{R}_E$ is denoted by the vector $v = (\dot{x}, \dot{y}, \dot{z})^T$ and is expressed as follows:

$$v = \mathscr{R}(\phi, \theta, \psi).v_B \quad (8)$$

where v_B is the linear velocity of the Quadrotor in the body attached frame.

Let us consider m as the total mass of the Quadrotor, g represents the gravity and l the distance from the center of each rotor to the CoG Mederreg et al. [2004].

Consider the vector $\vartheta = (p, q, r)^T$ which denotes the angular velocity in the frame $\mathbf{R}_B$. This vector can be transformed from the body frame $\mathbf{R}_B$ into the inertial one $\mathbf{R}_E$ as follows:

$$\vartheta = \begin{pmatrix} \dot{\phi} - s\theta\dot{\psi} \\ c\phi\dot{\theta} + s\phi c\theta\dot{\psi} \\ c\phi c\theta\dot{\psi} - s\phi\dot{\theta} \end{pmatrix} \quad (9)$$

So, we can deduce the angular velocities in the inertial frame which are given by the following transformation relationship:

$$\vartheta = \begin{pmatrix} p \\ q \\ r \end{pmatrix} = \begin{pmatrix} 1 & 0 & -s\theta \\ 0 & c\phi & s\phi c\theta \\ 0 & -s\phi & c\phi c\theta \end{pmatrix} \begin{pmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{pmatrix} = T(\phi, \theta)\dot{\eta} \quad (10)$$

where T is the well known velocities' transformation matrix which is invertible. $\dot{\eta} = [\dot{\phi}, \dot{\theta}, \dot{\psi}]^T$

2.4. Modelling with Newton-Euler formalism

The variations of the propellers rotation speeds produce gyroscopic torques. There are two rotational motions of the Quadrotor body:

$$\mathbf{M}_{gp} = \sum_{i=1}^{4} \Omega \wedge \left(0, 0, J_r(-1)^{i+1}\omega_i\right)^T \quad (11)$$

$$\mathbf{M}_{gb} = \Omega \wedge J\Omega \quad (12)$$

where Ω is the angular velocities vector in the fixed-frame, J_r is the propeller inertia for each rotor. The inertia matrix J of the Quadrotor body is defined as follows:

$$J = \begin{pmatrix} J_{xx} & 0 & 0 \\ 0 & J_{yy} & 0 \\ 0 & 0 & J_{zz} \end{pmatrix} \quad (13)$$

Using the Newton-Euler formalism for modelling, the Newton's laws lead to the following motion equations of the Quadrotor:

$$\begin{cases} m\ddot{\xi} = \mathbf{F}_{th} - \mathbf{F}_d + \mathbf{F}_g \\ J\Omega = \mathbf{M} - \mathbf{M}_{gp} - \mathbf{M}_{gb} - \mathbf{M}_a \end{cases} \quad (14)$$

where $\mathbf{F}_{th} = R(\phi, \theta, \psi)(0, 0, F)^T$ denotes the total thrust force of the four rotors, $\mathbf{F}_d = diag(\kappa_1, \kappa_2, \kappa_3)v_e^T$ is the air drag force which resists to the Quadrotor motion, $\mathbf{F}_g = (0, 0, mg)^T$ is the gravity force, $\mathbf{M} = \left(\Gamma_\phi, \Gamma_\theta, \Gamma_\psi\right)^T$ represents the total rolling, pitching and yawing torques. The terms $\mathbf{M}_{gp}$ and $\mathbf{M}_{gb}$ are the gyroscopic torques and $\mathbf{M}_a = diag(\kappa_4, \kappa_5, \kappa_6)\vartheta^T$ is the torque resulting from aerodynamic frictions.

By substituting the position vector and the forces with their expressions into Eq. (14), we have the following translation dynamics of the Quadrotor:

$$\begin{cases} \ddot{X} = & -\frac{\kappa_1}{m}\dot{X} + \frac{1}{m}(c\phi c\psi s\theta + s\phi s\psi).u_1 \\ \ddot{Y} = & -\frac{\kappa_2}{m}\dot{Y} + \frac{1}{m}(c\phi s\psi s\theta - s\phi c\psi).u_1 \\ \ddot{Z} = & \frac{1}{m}c\phi c\theta u_1 - g - \frac{\kappa_3}{m}\dot{Z} \end{cases} \quad (15)$$

From the second part of Eq. (14), and while substituting each moment by its expression, we deduce the following rotational dynamics of the rotorcraft:

$$\begin{cases} \dot{p} = & \frac{J_{yy} - J_{zz}}{J_{xx}}.q.r - \frac{J_r}{J_{xx}}\omega_r.q - \frac{\kappa_4}{J_{xx}}.p + \frac{1}{J_{xx}}.u_2 \\ \dot{q} = & \frac{J_{zz} - J_{xx}}{J_{yy}}.p.r + \frac{J_r}{J_{yy}}\omega_r.p - \frac{\kappa_5}{J_{yy}}q + \frac{1}{J_{yy}}.u_3 \\ \dot{r} = & \frac{J_{xx} - J_{yy}}{J_{zz}}.p.q - \frac{\kappa_6}{J_{zz}}r + \frac{1}{J_{zz}}.u_4 \end{cases} \quad (16)$$

According to the established equations (10), (15) and (16), $x = \left(X, \dot{X}, Y, \dot{Y}, Z, \dot{Z}, \phi, p, \theta, q, \psi, r\right)$ is retained as the state-space vector of the nonlinear model of the

Quadrotor rewritten as the following form $\dot{x} = f(x,u)$:

$$\dot{x} = \begin{cases} \dot{x}_1 = & x_2 \\ \dot{x}_2 = & a_9 x_2 + \frac{1}{m}(c(x_7)c(x_9)s(x_{11}) + s(x_7)s(x_1 1))u_1 \\ \dot{x}_3 = & x_4 \\ \dot{x}_4 = & a_{10}x_4 + \frac{1}{m}(c(x_7)s(x_9)s(x_{11}) - s(x_7)c(x_{11}))u_1 \\ \dot{x}_5 = & x_6 \\ \dot{x}_6 = & a_{11}x_6 + \frac{c(x_7)c(x_9)}{m}u_1 - g \\ \dot{x}_7 = & x_8 + x_{10}\, s(x_7)\tan(x_9) + x_{12}\, c(x_7)\tan(x_9) \\ \dot{x}_8 = & a_1 x_{10}x_{12} + a_2 x_8 + a_3\omega_r x_{10} + b_1 u_2 \\ \dot{x}_9 = & x_{10}\, c(x_7) - x_{12}\, s(x_7) \\ \dot{x}_{10} = & a_4 x_8 x_{12} + a_5 x_{10} + a_6\omega_r x_8 + b_2 u_3 \\ \dot{x}_{11} = & x_{10}\, s(x_7)\, sec(x_9) + x_{12}\, c(x_7)\, sec(x_9) \\ \dot{x}_{12} = & a_7 x_8 x_{10} + a_8 x_{12} + b_3 u_4 \end{cases} \tag{17}$$

where $a_1 = (J_y - J_z)/J_x$, $a_2 = -\kappa_4/J_x$, $a_3 = -J_r/J_x$, $a_4 = (I_z - I_x)/J_y$, $a_5 = -\kappa_5/J_y$, $a_6 = -J_r/J_y$, $a_7 = (I_y - I_x)/J_z$, $a_8 = -\kappa_6/J_z$, $a_9 = -\kappa_1/m$, $a_{10} = -\kappa_2/m$, $a_{11} = -\kappa_3/m$, $b_1 = -l/J_x$, $b_2 = -l/J_y$, $b_3 = -l/J_z$,
Note that $\kappa_{1,2,\ldots,6}$ are the aerodynamic friction and translational drag coefficients, $\omega_r = \omega_1 - \omega_2 + \omega_3 - \omega_4$ is the overall residual rotor angular speed.

2.5. Actuators Dynamics

The created aerodynamical torques and forces, given in Eq. (1) to Eq. (5), respectively, are provided through four Brushless DC motors which are characterize by a high torque and little friction (Austin [2010]). We consider that these motors have the same behavior of conventional DC motor at the steady state regime. Hence, the armature voltage of the ith Brushless DC motor is defined as follows:

$$v_i = \frac{R_{mot}}{k_{mot}} J_r \dot{\omega}_i + k_{mot}\omega_i + dR_{mot}\omega_i^2 \tag{18}$$

where R_{mot} and k_{mot} denote the internal resistance and torque coefficient of the Brushless motors, respectively, d is the drag propellers' coefficient.
Since that the drag coefficient d is very small, this dynamic can be approximated as a first order lag transfer function where the characteristic parameters can be identified by experimental trials as shown in (Becker et al. [2012]).
So, the Quadrotor is controlled by varying the force generated by each motor, the other terms, by varying the motors speed.

2.6. Sensors Dynamics

The Quadrotor states are measured using an Inertial Measurement Unit (IMU) which contains accelerometers and gyroscope sensors (Guerrero and Lozano [2012], Becker et al. [2012], Austin [2010]). These give us measurements of the translational and rotational velocities. The translation and rotation outputs measurements along x, y, and z axes can be described by Eq. (19) and Eq. (20) respectively.

$$y^{acc} = \alpha^{acc} v_B + \beta^{acc} + \gamma^{acc} \tag{19}$$

$$y^{gyro} = \alpha^{gyro}\vartheta + \beta^{gyro} + \gamma^{gyro} \tag{20}$$

where $y^{acc} = (y_X^{acc}, y_Y^{acc}, y_Z^{acc})^T$ are the sensor outputs, $\alpha^{acc} = diag\,(\alpha_X^{acc}, \alpha_Y^{acc}, \alpha_Z^{acc})$ denote the accelerometer gains, $v_B = \mathscr{R}^{-1}(\phi,\theta,\psi)\,v$ presents the linear velocities in the body-frame, $\beta^{acc} = (\beta_X^{acc}, \beta_Y^{acc}, \beta_Z^{acc})^T$ are the sensor bias and $\gamma^{acc} = (\gamma_X^{acc}, \gamma_Y^{acc}, \gamma_Z^{acc})^T$ present the zero mean white noises.
where $y^{gyro} = \left(y_X^{gyro}, y_Y^{gyro}, y_Z^{gyro}\right)^T$ are the sensor outputs' voltage, $\alpha^{gyro} = diag\left(\alpha_X^{gyro}, \alpha_Y^{gyro}, \alpha_Z^{gyro}\right)$ are the gyroscope gains, $\vartheta = (p,q,r)^T$ denotes the angular velocities in the body-frame, $\beta^{gyro} = \left(\beta_X^{gyro}, \beta_Y^{gyro}, \beta_Z^{gyro}\right)^T$ are the sensor bias and $\gamma^{gyro} = \left(\gamma_X^{gyro}, \gamma_Y^{gyro}, \gamma_Z^{gyro}\right)^T$ present the zero mean white noises.

3. CONTROL STRATEGY

The Quadrotor control strategy can be split in four blocks or steps in the control loop. The first step (motors block) consists in controlling each motor separately to produce the propelling forces. Unlike the commonly used methods which control the angular speed of motors, here, we use the generated thrust F_i as the output force to be controlled from corresponding voltage input v_i of the motors (see Fig (3)).
In another hand, the Quadrotor contains four inputs u_i and six output position variables to be controlled, thus it is not possible to control all states separately at the same time. Generally, to overcome this problem, the procedure consists of controlling the cartesian positions x, y, z and the drone orientations angle ψ. The remaining two angles ϕ and θ are internal states which are not directly controlled. Their reference signals are deduced from the outputs in order to control the longitudinal and lateral directions, respectively x, y. Then, ϕ and θ, are controlled by an inner loop form the second step. Then x and y are controlled in the third step. The control of z and ψ is done directly in the third block.
The trajectories to be followed are generated by using IBVS approach, this defines the last control step. In the Fig. 2 we summarize the proposed control strategy of the whole system. This avoids the use of a joystick (or the human in the loop).

3.1. Actuator controllers

The most common motor controllers in the Quadrotor literature use the speed motors ω_i as reference signal to be controlled from input v_i, then the forces $F_i = \omega_i^2$ are generated. The speed output signal should be squared before being used as the force control signal (in the second step) since the thrust is assumed only proportional to the square of the motor speeds. This induces neglected dynamics in the propelling forces. This can be seen as inherent control perturbation.
Here, we propose that the forces F_i take place of the controlled signal instead of the motors speed ω_i. Then a simple Proportional Integral (PI) controller is widely efficient with a very small time response and zero steady

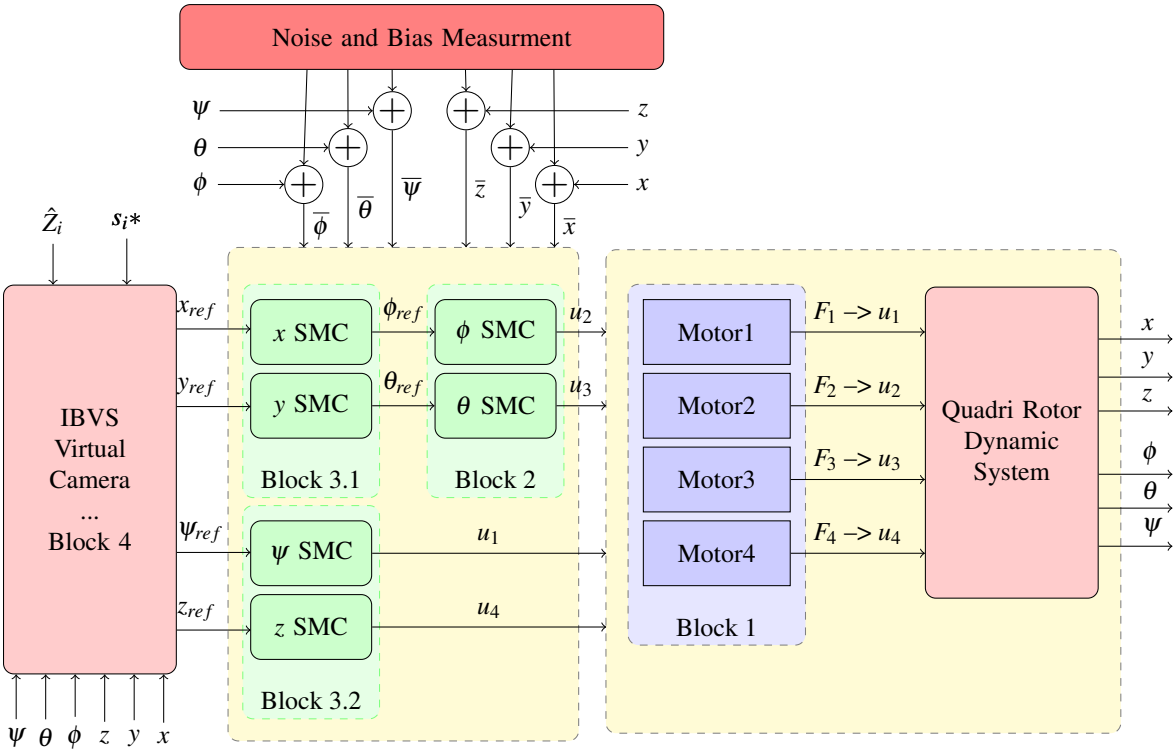

Figure 2: Control strategy of SMC using IBVS for Quadrotor

state error to track the desired propelling forces. The Fig. 3 depicts the scheme of the proposed motor control.

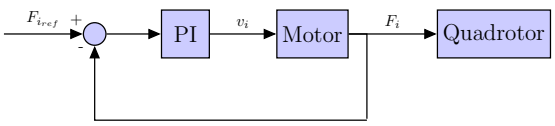

Figure 3: Motor control loop

3.2. Sliding Mode Control

The sliding mode control achieves the desired configuration in two steps M'Sirdi and Nadjar-Gauthier [2002]. The first is to drag all states toward this desired configuration called sliding surface $S(e,t)$ and enforce them to manifest around it Mederreg et al. [2005], Mokhtari et al. [2006]. Generally, the sliding surface of first order is given by:

$$S(e,t) = \dot{e} + \lambda e \tag{21}$$

where $e = x - x_d$ and λ is positive constant coefficient. The stabilizing control law leading to $S(e,t) = 0$ is deduced by using the Lyapunov function defined as follows:

$$V(e) = \frac{1}{2} S(e,t)^2 \tag{22}$$

As previously indicated, the control law is formed by two terms; the equivalent control law and the switching control law:

$$u = u_{eq} + u_s \tag{23}$$

The equivalent part is a continuous control law deduced from $\frac{\partial S(e,t)}{\partial t} = \dot{S}(e,t) = 0$ (using an available approximate model) and the second control part u_s has a discontinuous feature defined as in M'Sirdi and Nadjar-Gauthier [2002]:

$$u_s = -K\, sign(S(e,t)) \tag{24}$$

where K is a positive constant and *sign* is the sign function.

To design SMC for Quadrotor, we use an approximate model, rather than the defined one in Eq.17 for simulation, which parameters are not well known (parametric uncertainty).

$$\dot{\bar{x}} = \bar{f}(\bar{x},u) : \begin{cases} \dot{\bar{x}}_1 = \bar{x}_2 \\ \dot{\bar{x}}_2 = \bar{a}_9 \bar{x}_2 + \frac{1}{m} u_x u_1 \\ \dot{\bar{x}}_3 = \bar{x}_4 \\ \dot{\bar{x}}_4 = \bar{a}_{10} \bar{x}_4 + \frac{1}{m} u_y u_1 \\ \dot{\bar{x}}_5 = \bar{x}_6 \\ \dot{\bar{x}}_6 = \bar{a}_{11} x_6 + \frac{c(\bar{x}_7)c(\bar{x}_9)}{m} u_1 - g \\ \dot{\bar{x}}_7 = \bar{x}_8 \\ \dot{\bar{x}}_8 = \bar{a}_1 \bar{x}_{10} \bar{x}_{12} + \bar{a}_2 \bar{x}_8 + \bar{a}_3 \omega_r \bar{x}_{10} + \bar{b}_1 u_2 \\ \dot{\bar{x}}_9 = \bar{x}_{10} \\ \dot{\bar{x}}_{10} = \bar{a}_4 \bar{x}_8 \bar{x}_{12} + \bar{a}_5 \bar{x}_{10} + \bar{a}_6 \omega_r \bar{x}_8 + \bar{b}_2 u_3 \\ \dot{\bar{x}}_{11} = \bar{x}_{10} \\ \dot{\bar{x}}_{12} = \bar{a}_7 \bar{x}_8 \bar{x}_{10} + \bar{a}_8 \bar{x}_{12} + \bar{b}_3 u_4 \end{cases} \tag{25}$$

where $\bar{x}_i$ are biased measurements perturbed by noise, $(\bar{a}_i, \bar{b}_i)$ are the approximated model parameters:

$$(\bar{a}_i, \bar{b}_i) = (a_i, b_i) \pm 25\%(a_i, b_i)$$

SMC in altitude and attitude control uses u_i as control input to track the desired sliding surface. For example, the sliding surface as defined in Eq.21 for the altitude is given as:

$$S_z = \bar{x}_6 - x_{d_6} - \lambda_z(\bar{x}_5 - x_{d_5}) \tag{26}$$

Then the equivalent control u_{eq} is obtained from the condition:

$$\begin{aligned} \dot{S}_z = 0 \quad &\Leftrightarrow \bar{a}_{11}\bar{x}_6 + \tfrac{c(\bar{x}_7)c(\bar{x}_9)}{m}u_1 - g - \dot{x}_{d_6} + \lambda_z(\bar{x}_6 - x_{d_6}) = 0 \\ &u_{eq_z} = \tfrac{m}{c(\bar{x}_7)c(\bar{x}_9)}(g + \dot{x}_{d_6} - a_{11}\bar{x}_6 - \lambda_z(\bar{x}_6 - x_{d_6})) \end{aligned} \tag{27}$$

The VSS (Variable Structure Control) control is given by M'Sirdi and Nadjar-Gauthier [2002]:

$$u_{s_z} = -k_z sign(S_z) \tag{28}$$

The complete control law can be written according to Eq.23:

$$u_1 = \frac{m}{c(\bar{x}_7)c(\bar{x}_9)}(g + \dot{x}_{6_d} - a_{11}\bar{x}_6 - \lambda_z(\bar{x}_6 - x_{6_d})) - k_z sign(S_z)$$

The same steps are followed to extract other control laws:

$$\begin{aligned} u_2 = \quad & \tfrac{1}{\bar{b}_1}(\dot{x}_{8_d} - \bar{a}_1\bar{x}_{10}\bar{x}_{12} - \bar{a}_2\bar{x}_8 - \bar{a}_3\omega_r\bar{x}_{10} - \lambda_\phi(\bar{x}_8 - x_{8_d})) \\ & -k_\phi sign(S_\phi) \end{aligned}$$

$$\begin{aligned} u_3 = \quad & \tfrac{1}{\bar{b}_2}(\dot{x}_{10_d} - \bar{a}_4\bar{x}_8\bar{x}_{12} - \bar{a}_5\bar{x}_{10} - \bar{a}_6\omega_r\bar{x}_8 - \lambda_\theta(\bar{x}_{10} - x_{10_d})) \\ & -k_\theta sign(S_\theta) \end{aligned}$$

$$u_4 = \frac{1}{\bar{b}_3}(\dot{x}_{12_d} - \bar{a}_7\bar{x}_8\bar{x}_{10} - \bar{a}_8\bar{x}_{12} - \lambda_\psi(\bar{x}_{12} - x_{12_d})) - k_\psi sign(S_\psi)$$

The x, y directions are controlled by the virtual control law u_x and u_y, respectively.

$$u_x = \frac{m}{u_1}(\dot{x}_{2_d} - \bar{a}_9\bar{x}_2 - \lambda_x) - k_x sign(S_x)$$

$$u_y = \frac{m}{u_1}(\dot{y}_{4_d} - \bar{a}_{10}\bar{x}_4 - \lambda_y) - k_y sign(S_y)$$

Unlike x_d, y_d, z_d, ψ_d which are derived from the visual servoing task, ϕ_d and θ_d are obtained from both virtual control laws u_x, u_y as follows:

$$\begin{bmatrix} \phi_d \\ \theta_d \end{bmatrix} = \begin{bmatrix} \sin(\psi_d) & \cos(\psi_d) \\ -\cos(\psi_d) & \sin(\psi_d) \end{bmatrix}^{-1} \begin{bmatrix} u_x \\ u_y \end{bmatrix} \tag{29}$$

4. IBVS CONTROLLER

IBVS is a controller essentially based on visual data extracted from a camera. This approach has been emerged form Task Oriented Feedback Approach (see Samson and Espiau [1990] in the aim of controlling robot manipulators. This is called Eye-in-hand or Eye-to-hand, depending to the location of the camera with respect to robot Hutchinson et al. [1996]. In Vision Servoing, the control strategy is based on a cost function minimization Chaumette and Hutchinson [2006].
As task function cost we can take:

$$e_i(t) = s_i(t) - s_i^* \tag{30}$$

where $s_i(t)$ and s_i^* are the image measurement and desired configuration in the image plane, respectively. These measurements are defined in pixels $s_i = (p_{x_i}, p_{y_i})^T$ and they are used to determine the corresponding point in image frame, denoted by $p_i = (x_i, y_i)$. Relationship between both points is defined by applying the inverse of the camera intrinsic matrix:

$$\begin{bmatrix} x_i \\ y_i \\ 1 \end{bmatrix} = \begin{bmatrix} f_{xx} & f_{xx}\alpha & x_0 \\ 0 & f_{yy} & y_0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} p_{x_i} \\ p_{y_i} \\ 1 \end{bmatrix} \tag{31}$$

In the Eye-in-hand approach adopted in this work, the relationship between the projection measurement in the image frame p_i and the target point defined in the camera frame, denoted by $P_{c_i} = (X_{c_i}, Y_{c_i}, Z_{c_i})$ is derived from the Pinhole camera model:

$$\begin{cases} x_i = \frac{X_{c_i}}{Z_{c_i}} \\ y_i = \frac{Y_{c_i}}{Z_{c_i}} \end{cases} \tag{32}$$

When the camera is moving with $V_c = [\upsilon_c, \omega_c]^T = [\upsilon_x, \upsilon_y, \upsilon_z, \omega_x, \omega_y, \omega_z]^T$, the dynamics of P_{c_i} are given by applying the following formula Chaumette and Hutchinson [2006]:

$$\dot{P}_{c_i} = \upsilon_{c_i} + \omega_{c_i} \times P_{c_i} \tag{33}$$

with $\times$ is the cross product. The dynamics of the point in image plane is taken from the derivative of Eq.32:

$$\begin{cases} x_i = \frac{\dot{X}_{c_i}}{Z_{c_i}} - \frac{X_{c_i}\dot{Z}_{c_i}}{Z_{c_i}^2} \\ y_i = \frac{\dot{Y}_{c_i}}{Z_{c_i}} - \frac{Y_{c_i}\dot{Z}_{c_i}}{Z_{c_i}^2} \end{cases} \tag{34}$$

Using Eq.33 and Eq.34, the following results are obtained:

$$\begin{bmatrix} \dot{x}_i \\ \dot{y}_i \end{bmatrix} = \mathbb{L} \begin{bmatrix} \upsilon_c \\ \omega_c \end{bmatrix} \tag{35}$$

where $\mathbb{L}$ is the interaction matrix:

$$\mathbb{L} = \begin{bmatrix} \frac{1}{Z_{c_i}} & 0 & -\frac{x_i}{Z_{c_i}} & -x_i y_i & 1 + x_i^2 & -y_i \\ 0 & \frac{1}{Z_{c_i}} & -\frac{y_i}{Z_{c_i}} & -(1 + y_i^2) & x_i y_i & x_i \end{bmatrix} \tag{36}$$

It is clear, to compute all components of the camera velocity, we need more than three points with different depth. If the feature points to be tracked are co-planar, we need at least four points.

4.1. Trajectory generation based on visual tracking

The key component of visual servoing controller is the interaction matrix. The common approach considered in IBVS control problem is based on the choice of an exponential decreasing of the cost function:

$$\dot{e}_i = -\delta e_i(t) \tag{37}$$

where the decreasing rate δ is a positive constant. Under the assumption that the desired configuration is constant and using the Eq.35, the derivative of the error is:

$$\dot{s}(t) = -\delta e(t) = \mathbb{L}V_c \tag{38}$$

This leads to the camera velocity:

$$V_c = -\delta \mathbb{L}^{+} e(t) \tag{39}$$

where $\mathbb{L}^{+} \in \mathbb{R}^{6\times 2k}$ is a pseudo-inverse matrix with $k \geq 4$ as the image number of points to be tracked. If only one camera is used, is it not possible to define exactly the depth Z_{c_i}. Consequently, the interaction matrix is not well defined. To overcome this problem, this matrix can be estimated or approximated by fixing z_c to its value of the desired configuration.

4.2. IBVS using a virtual camera

It is expected that the IBVS controller generates six desired velocities based on the observed object. The trajectories are defined in the camera frame and should be transformed to the Quadrotor frame. Under assumption of any translation and rotation between virtual camera and Quadrotor frames, the generated trajectories are directly used by Quadrotor Controller.

However, since the pitch and roll angles in Quadrotor are implicitly used to drive the motion in x and y direction, ω_x and ω_y they are not addressed for tracking task. This might cause disappearance of the target from the view field of the camera. The proposed solution is to use saturation in pitch and roll variations. The final process to obtain Quadrotor references is defined as:

$$[\dot{X}_{ref}, \dot{Y}_{ref}, \dot{Z}_{ref}] = \mathscr{R}[\upsilon_x, \upsilon_y, \upsilon_z] \tag{40}$$

and for rotation:

$$[\dot{\phi}_{ref}, \dot{\theta}_{ref}, \dot{\psi}_{ref}] = T(\phi, \theta)[\omega_x, \omega_y, \omega_z] \tag{41}$$

5. SIMULATION RESULTS

In order to verify the performance of the control strategy, the simulations was developed under Matlab/Simulink software. Various simulations are conducted to achieve the final task. The Quadrotor and motors parameters used in all our simulation are reported in Table.1

The parameters of the sensor dynamic applied to all outputs are chosen as follows: $\alpha^{acc} = \alpha^{gyro} = 1$, $\beta^{acc} = \beta^{gyro} = 10^{-3}$. As for the noise, we have used the Simulink Band-Limited White Noise block generator with noise power equal to 10^{-4}.

For the visual servoing task, intrinsic parameters of the virtual camera are inspired from a real camera parameters; $(f_{xx}, f_{yy}) = (657.4, 657.8)$, $\alpha = 10^{-4}$, $(x_0, y_0) = (303, 243)$. During the simulation, the depth Z_{c_i} defined in interaction matrix was fixed to be equal to $z_{fd} = 0.8$.

First, the trajectory generation based on visual servoing and SMC control for Quadrotor were tested separately (see figures 4). Under the assumption the camera frame superposes the Quadrotor frame, the desired target in image plane corresponds to the desired final pose $([x_{fd}, y_{fd}, z_{fd}d, \phi_{fd}, \theta_{fd}, \psi_{fd}] = [0, 0, 0.8, 0, 0, \frac{pi}{4}])$. The evolution of the virtual camera frame and the generated trajectories using IBVS without Quadrotor is shown in Fig.4. The considered target is a plane with four points.

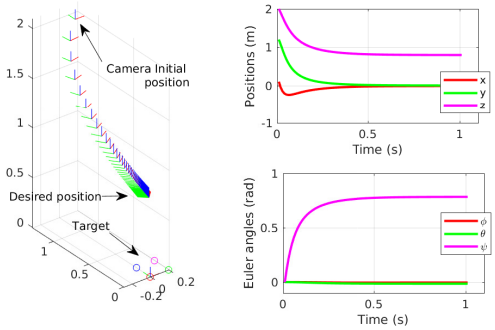

Figure 4: Virtual camera frame evolution in IBVS generation trajectories without Quadrotor

Now, we apply these trajectories to Quadrotor using SMC. For all the simulation tests, the SMC control is based on an approximate model (with less dynamics) and with uncertainties in all the a priori estimated parameters. After many tests, the SMC gain parameters are selected to be $k_x = k_y = 0.9$, $k_z = k_\phi = k_\theta = 0.5$ and $k_\psi = 0.01$.

The visual servoing trajectory generation is supposed launched from the following initial pose $[x_0, y_0, z_0, \phi_0, \theta_0, \psi_0] = [0.3, -0.3, 1.2, 0, 0, 0]$. However, this pose is different to take-off pose of Quadrotor. Then, before launching the visual servoing task, the Quadrotor performs the path between both poses. In simulation, this step takes the first $20s$.

Simulation results with a sliding mode controller by performing a visual tracking are shown in Fig.5 and Fig.6 which shows the positions and the Euler anglers, respec-

Parameter	Value	Parameter	Value
m	0.486	g	0.486
l	0.25	b	2.9842e-5
d	3.2320e-7	J_x	2.8385e-5
J_y	2.8385e-5	J_z	2.8385e-5
κ_1	5.5670e-4	κ_2	5.5670e-4
κ_3	6.3540e-4	κ_4	5.5670e-4
κ_5	5.5670e-4	κ_6	6.3540e-4
R_{mot}	6.3540e-4	J_r	2.8385e-5
k_{mot}	$20J_r$		

Table 1: Quadrotor parameters values

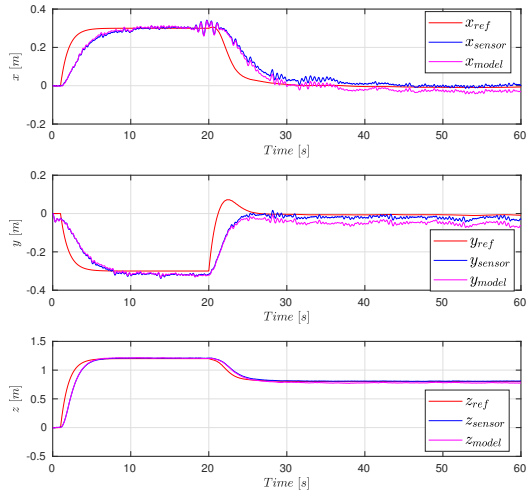

Figure 5: Time evolution of reference trajectories, outputs issue from sensors and model outputs in x,y and z directions

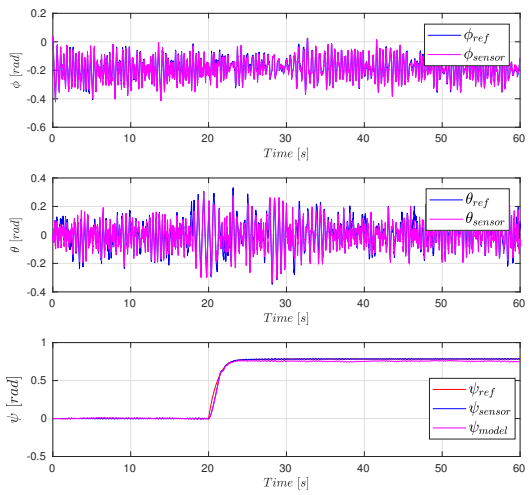

Figure 6: Time evolution of reference trajectories, outputs issue from sensors for ϕ and θ angles. In addition, output model for ψ

tively. The effect of the sensors dynamic and perturbation on the outputs of Quadrotor can be seen clearly.
Despite, the proposed approach shows a considerable ability to reject the perturbations. Since the yaw angle and the altitude are controlled directly, they achieve the desired reference in shorter time with less oscillations with respect to x and y directions. This is mainly due to the strong coupling effects between the blocks in the control scheme of Fig.2.
In addition, the peak shown is time evolution in y direction is generated by convergence rate coefficient δ in the visual servoing task. To reduce this effect, we can use a weighting matrix rather than using a scalar coefficient to damp more or less the convergence rate along this direction. Thus, instead of using δ, we use $\Delta_{6\times 6} = diag\{\delta_1, ..., \delta_6\}$. However, if δ_2 is selected to be so small, this can produce a slight static error in this direction.

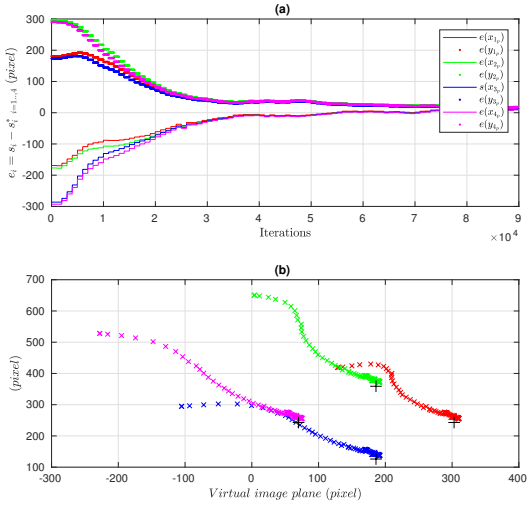

Figure 7: Tracking errors evolution of all points in image pixel measurement (a). Tracking evolution in image plane (b)

A well known drawback of SMC is the chattering effects, which appears in roll and pitch time evolution. The convergence criteria can be also evaluated based on the visual data. Fig.7.a shows the error evolution of each coordinate of all points and Fig.7.b is the evolution of each point in the image plane of the virtual camera. Both figures show an error equal to ± 10 *pixels* which is considered to be within an acceptable variation range. Basically, this error is caused by the measurement bias of the sensors.

Through the last simulation shown in Fig.8, we try to illustrate the effectiveness of our approach when the target is moving. Here, the center of target performs a circular motion in (x, y) plan. Despite that the Quadrotor follows the circular motion, there is a small delay in time response. This is due to neglected term of $\frac{\partial s(t)^*}{\partial t}$ in task function which becomes different to zero.

6. CONCLUSION

This paper proposes a robust control strategy for a Quadrotor UAV system endowed with a virtual camera.
To simulate all effects that can influence the evolution of the Quadrotor in real environment, we have developed a dynamic model taking into account physical parts and aerodynamics phenomena, as well as different hardware components such as the actuators with their controller (force control) and sensors dynamics and perturbations.
In simulations, we remark that bias of sensors cause gaps between model outputs and sensors measurements. To overcome the nonlinear effects and for robustness, a sliding mode control has been developed and tested with use of the IBVS. Here, the SMC uses an approximate dynamic (with less dynamics) in the control model and uncertainties in all the control model parameters.
To verify the robustness of the proposed controller, the

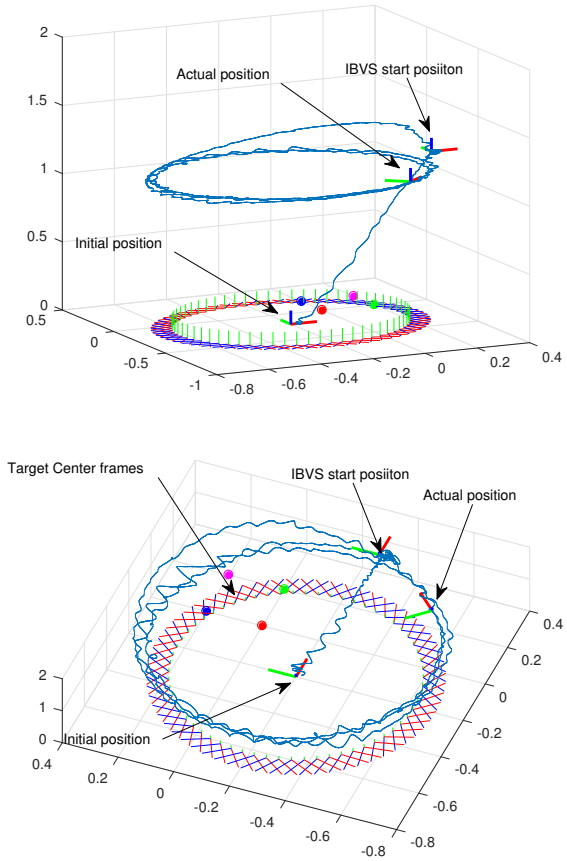

Figure 8: Tracking behavior for circular trajectory

Quadrotor tracks trajectories generated by visual servoing task using a virtual camera. The obtained simulation results show the efficiency of the proposed controller to track the generated trajectories in spite of the complexity and the nonlinearities of the whole system.
The proposed control approach is also tested with a moving target for evaluation of the tracking capabilities. As prospects, we hope to reduce the shattering effect by adopting a smooth sign function or by using sliding mode control with a higher order. As for the engendered error in image plane, we hope to combine sensor measurements with the optical flow from image information to well estimate the Quadrotor movements.

REFERENCES

R. Austin. *Unmanned Aircraft Systems: UAVs design, development and deployment*. 1st edition, John Wiley and Sons Ltd, London, 2010.

S. Azrad, F. Kendoul, and K. Nonami. Visual servoing of quadrotor micro-air vehicle using color-based tracking algorithm. *J. Syst. Des. Dyn.*, 4(2), 2010.

Y Zhang B. Wang, L. Mu. Adaptive robust tracking control of quadrotor helicopter with parametric uncertainty and external disturbance. In *International Conference on Unmanned Aircraft Systems (ICUAS)*, 2017.

M. Becker, B. Coronel, R. Sampaio, S. Bouabdallah, V. De Perrot, and R. Siegwart. In flight collision avoidance for a mini-uav robot based on onboard sensors. *J. of the Brazilian Society of Mechanical Sciences and Engineering*, 34(3):294–307, 2012.

S. Bouabdallah. *Design and control of quadrotors with application to autonomous flying*. PhD thesis, École Polytechnique fédérale de Lausanne, 2007.

H.L. Chan and K.T. Woo. Design and control of small quadcopter system with motor closed loop speed control. *International Journal of Mechanical Engineering and Robotics Research*, 4(3), 2015.

F. Chaumette and S. Hutchinson. Visual servo control part i: basic approaches. *IEEE Robot. Autom. Mag*, 13 (4), 2006.

C. Cosmin and J. B. Macnab. A new robust adaptive-fuzzy control method addaplied to quadrotor helicopter stabilization. In *Fuzzy Information Processing Society, Annual Meeting of the North American, NAFIPS*, pages 454–458, 2006.

A. Das, F. Lewis, and S. Subbarao. Dynamic neural network based robust backstepping control approach for quadrotors. In *Proc. of the AIAA Guidance, Navigation and Control Conference and Exibit , Hawaii, USA*, 2008.

A. Das, K. Subbarao, and F. Lewis. Dynamic inversion with zero-dynamics stabilisation for quadrotor control. *IET Control Theory Application*, 3(3):303–314, 2009.

S.V. Emelyanov. Theory of variable-structure control systems: inception and initial development. *Computational Mathematics and Modeling*, 18(4), 2007.

V. Grabe, H.H. Bulthoff, and P.R. Giordano. A comparison of scale estimation schemes for a quadrotor uav based on optical flow and imu measurements. In *IEEE - RSJ Int. Conf. on Intelligent Robots and Systems, IROS*, 2013.

A. Guerrero and R. Lozano. *Flight formation control*. John Wiley and Sons Inc, New York, 2012.

T. Hamel and R. Mahony. Visual servoing of an underactuated dynamic rigid-body system: an image-based approach. *IEEE Trans. Robot. Autom*, 18(2), 2006.

B. Herisse, T. Hamel, R. Mahony, and F.X. Russotto. Landing a vtol unmanned aerial vehicle on a moving platform using optical flow. *IEEE Trans. Robot*, 28(1), 2012.

S. Hutchinson, G.D. Hager, and P.I. Corke. A tutorial on visual servo control. *IEEE TRANSACTIONS ON ROBOTICS AND AUTOMATION*, 12(5), 1996.

S. Islam, P. X. Liu, and A. El Saddik. Robust control of four-rotor unmanned aerial vehicle with disturbance

uncertainty. *IEEE Transactions on Industrial Electronics*, 62(3):1563–1571, March 2015. ISSN 0278-0046. doi: 10.1109/TIE.2014.2365441.

R. Lozano. *Unmanned aerial vehicles: Embedded control*. John Wiley and sons, 2013.

R. Mahony, P. Corke, and T. Hamel. Dynamic image-based visual servo control using centroid and optic flow features. *J. Dyn.Syst-T. ASME*, 130, 2017.

N. Manamanni, M. Djemai, T. Boukhobza, and N.K. M'Sirdi. Nonlinear sliding observer based control for a pneumatic robot leg. *International Journal of Robotics and Automation*, 16:100–112, 01 2001.

L. Mederreg, F. Diaz, and N.K. M'Sirdi. Nonlinear backstepping control with observer design for 4 rotors helicopter. In *Proceedings of AVCS 2004*, Genova, 2004.

L. Mederreg, F. Diaz, and N.K. M'Sirdi. Dynamic feedback control for a quadrotor unmanned aerial vehicle. In *SSD 2005*, Sousse, Tunisia, 2005.

V. Mistler, A. Benallegue, and N.K. M'Sirdi. Exact linearization and noninteracting control of a 4-rotors helicopter via dynamic feedback. In *ROMAN 10th IEEE Int. Workshop on Robot-Human Interactive Communication, Bordeaux*, pages 586–593, Bordeaux and Paris, 2001. ISBN 0-7803-7222-0. doi: 10.1109/ROMAN.2001.981968.

V. Mistler, A. Benallegue, and N.K. M'Sirdi. Linéarisation exacte et découplage entrées-sorties, comparaison entre l'hélicoptère standard et l'hélicoptère 4 rotors. In *Proceedings of the CIFA 2002*, Nantes, 2002.

A. Mokhtari, N.K. M'Sirdi, K. Meghriche, and A Belaidi. Feedback linearization and linear observer for a quadrotor unmanned aerial vehicle. *Advanced Robotics*, 20(1):71–91, 2006. doi: DOI:10.1163/156855306775275495. URL https://doi.org/10.1163/156855306775275495.

N.K M'Sirdi and N. Nadjar-Gauthier. *Application of Sliding Mode Control to Robotic Systems*, chapter 13, pages 351–387. Control Engineering Series. Marcel Dekker edited by Wilfrid Perruquetti and Jean Pierre Barbot, New York, 2002. URL https://books.google.fr/books?isbn=0203910850.

N.K. M'Sirdi, P. Fraisse, P. Dauchez, and N. Manamani. Sliding mode control for a hydraulic underwater manipulator. In *Syroco'97*, 1997.

A. Rabhi, M. Chadli, and C. Pegard. Robust fuzzy control for stabilization of a quadrotor. In *International Conference on Advanced Robotics*, pages 471–475, 2011.

G.V. Raffo, M. G Ortega, and F. R. Rubio. Nonlinear hinfinity controller for the quad-rotor helicopter with input coupling*. *IFAC Proceedings Volumes*, 44 (1):13834 – 13839, 2011. ISSN 1474-6670. doi: https://doi.org/10.3182/20110828-6-IT-1002.02453. URL http://www.sciencedirect.com/science/article/pii/S1474667016458477. 18th IFAC World Congress.

C. Samson and B. Espiau. Application of the task-function approach to sensor-based control of robot manipulators. *IFAC Proceedings Volumes*, 23 (8, Part 5):269 – 274, 1990. ISSN 1474-6670. doi: https://doi.org/10.1016/S1474-6670(17)51746-2. URL http://www.sciencedirect.com/science/article/pii/S1474667017517462. 11th IFAC World Congress on Automatic Control, Tallinn, 1990 - Volume 5, Tallinn, Finland.

H. Xie, K.H. Low, and Z. He. Adaptive visual servoing of unmanned aerial vehicles in gps-denied environments. *IEEE/ASME Trans.Mechatron*, 22(6), 2017.

ACKNOWLEDGMENT

This work is supported by the SASV research group and its funding. The research project was initiated and driven by N.K. M'Sirdi several years ago in the LRV (Robotics Laboratory of university of Versailles Saint Quentin). It is now, for SASV in a collaboration with the LAT of Tlemcen and other partners.

AUTHORS BIOGRAPHY

Choukri Bensalah is a post Doc at the LIS research laboratory of the University of Aix Marseille University, (LIS UMR CNRS 7020). His is assistant professor at University of Tlemcen, Algeria, he is member of "Laboratoire d'automatique de Tlemcen" since 2016. He was Graduated in "chaotic systems" from Tlemcen University (2007) and received the Ph.D from the Robotics Lab of the University Carlos III of Madrid (2014). His general research interests include modelling and simulation of dynamic systems, robotics and vision applications and control theory.

Nacer K. M'Sirdi is professor at Polytech Marseille and Aix Marseille University (AMU). He got the Phd in Electronics at ENSERG-INPG (1983) and the Doctorat d'Etat in adaptive signal processing at the ENSIEG-INPG, in 1988 (LAG). He was assistant professor, in University of Paris 6 in 1987 and Professor at University of Versailles in 1993. From 2005 up to now, he is a research member of the LIS (UMR CNRS 7020). In 2009 he has created the VSAS research project on Variable Structure Automatic Systems (SASV), for research in automatic control and optimization of VSS Systems with commutations.

Aziz Naamane is an Associate Professor at the Aix Marseille University, he is a member of Laboratoire d'Informatique et des Systmes (LIS UMR CNRS 7020). His main research activities deal with discrete event modelling and simulation, optimal control, diagnosis, vehicle dynamics and renewable energy. He is Member (co advisor) of the SASV (Variable Structure Automatic Systems) research Group and the HyRES Lab. His research field is now focused on Prototyping, Design and control of embedded power electric systems.