Comparative Study of Control Approaches Designed for a Quadrotor in a Visual Servoing Task without Observers

Choukri Bensalah, Nacer Kouider M'Sirdi, Aziz Naamane

Comparative Study of Control Approaches Designed for a Quadrotor in a Visual Servoing Task without Observers

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

***Abstract*— The performance and characteristics of four controllers applied to a quadrotor UAV are analyzed in this paper. The compared control approaches are: PID, Linearizing Feedback, Backstepping Approach and Sliding Mode based control. A suitable and efficient structure is proposed for control laws implementation. Standard sensors are used and need of observer is avoided by use of IBVS.**

The comparative analysis is done for the UAV in Visual Servoing Task. It is based on tracking errors during transients and at steady state, and on the performance and robustness with respect to plant model, to environment perturbations (in sensors and actuators) and uncertainties.

I. INTRODUCTION

Unmanned Aerial Vehicle (UAV) are nowadays very popular and multiple applications aroused (rescue, surveillance, inspection, mapping, bridges, buildings supervision, and cinema) [1]. They can perform stationary flight and take off and land vertically. Their manoeuvrability and controllability as well as their ability give them a key position to be used as autonomous mobile robots. They exhibit non linear, non-holonomic behaviour and, in their case, precise measurement of all pertinent signals is not possible either for cost, for feasibility or availability. However a human pilot is very often needed in the loop to compensates for the lack of sensors, drifts or robust estimators.

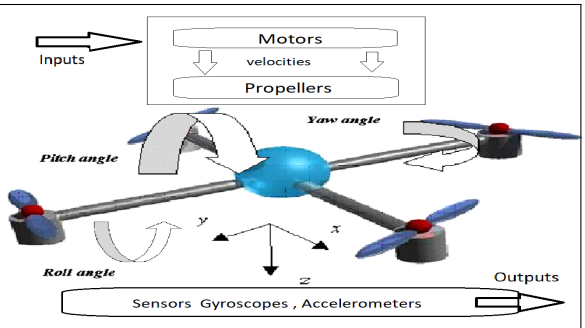

Fig. 1. Mechanical structure of the Quadrotor.

Our interest, in this paper, is focused in case of autonomous target tracking using control with embedded vision (instead of human in the loop) and use of standard sensors and no observers. The required application objective is capability to follow targets and realize autonomous motions. Then some features are required, like robustness face to uncertainties, measurement noise and to parameters variations and fast perturbations rejection.

The four rotors helicopter (see Fig 1) exhibits a nonlinear behavior which is subject to aerodynamic forces and moments and unknown disturbances. It is multi-variable, non linearly coupled and has inherent uncertainties in both high and low frequencies.

In literature, previous works have considered model analysis [1], [2] and have studied control design approaches suitable to this kind of systems [3], [4]. A Feedback Linearization approach have been developed and applied with and without use of a state observer [2], [5]. For such model, it has been shown that it is possible to apply a non linear controller with observers based on the Backstepping approach [3], [4]. This has shown a good robustness and permit to reduce the number of the required sensors to be used (using only gyro-meters and accelero-meters), by observing the absolute velocity of the helicopter which is difficult to measure. But the equations describing the altitude and the attitude motions of the Quadrotor helicopter are basically those of a rotating rigid body with six degrees of freedom (assuming perfect sensors and actuators).

It is true that non measurable signals can be obtained by successive differentiation. Unfortunately they will be contaminated by noise in a such a degree that they can no longer be used. This complicates further the design of robust controllers for actual systems. The main difficulties of the motion control, for high performance positioning, are parametric uncertainties, neglected dynamics, and external disturbances.

Several control approaches are applied and developed but very few of them take into account the sensors dynamics and the actuators transfer function, delay and dynamics, in presence of perturbations. It is not possible to get precise measurements of absolute position variables, without external sensors. The best way to track targets (without human in the loop) is to use vision sensors and this needs image data processing [6].

The main contribution of this paper is the use of complete dynamic model of UAV quadrotor, including sensors and actuators, with noise bias and dynamics. In addition, we propose a suitable and efficient structure for control laws implementation. This optimizes the controllability of the system. The need of observer to compensate for noise measurements, perturbations and drifts is avoided by use of IBVS and standard sensors.

Four control approaches, chosen for their important dif-

*This work was not supported by SASV group of LIS UMR CNRS 7020

[1]Nacer K. M'Sirdi and Aziz Naamane are with Aix Marseille University, Universit de Toulon, CNRS, LIS UMR 7020, SASV, Marseille, France nacer.msirdi@Lis-Lab.fr

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

ferences are developed and compared for their use in Image Vision Based Servoing for Target Tracking (**IBVS-TT**). Comparison between four control approaches is done: the Exact Feedback Linearization (**EFL**) with dynamic extension, Back-Stepping Control (**BSC**), Sliding Mode Control (**SMC**) and **PID**, are applied to a Quadrotor helicopter including sensors and actuators dynamics and compared.

This paper is structured as follows. The second section describes the quadrotor dynamics and model equations, for simulation, including the sensors and the actuators. The four designed controllers are presented in Section 3. The models and control validation and simulation results, in Matlab/Simulink, are discussed in section 4. This will be ended by a comparative analysis. Section 5 shows the performance of the selected controls when used with the IBVS-TT based approach. The last section concludes this paper.

II. UAV MODEL FOR SIMULATION AND CONTROL

The four rotors helicopter is propelled by four forces $F_{i(i\in\{1,2,3,4\})}$ produced by four DC motors $M_{i(i\in\{1,2,3,4\})}$ as depicted in Fig (1) [2]. 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 to control the altitude z for vertical take-off or landing. The velocity speeding up or slowing down the diagonal motors creates a moment which produces pitch motion ϕ. Pitch angle θ allows the Quadrotor to move towards the longitudinal direction x. Roll angle ϕ allows the Quadrotor to move towards the lateral direction y. Yaw angle ψ is obtained by speeding up or slowing down the clockwise motors regard to the others.

A. Actuators model

The forces are non linear function of $\omega_{1,2,3,4}$ the angular speeds of the four rotors and propellers, respectively. In general, they are assumed to be proportional to the square of the angular motors speeds ω_i and given by the equation

$$F_i = b\omega_i^2$$

where $b = \frac{1}{2}\rho\Lambda C_T r^2$ with ρ the air density, r and Λ are the radius and the section of the propeller, respectively. C_T is the aerodynamic thrust coefficient. The aerodynamic drag torques $\delta_i = d\omega_i^2$ produced at each actuator are opposed to the motor torque and proportional to the propeller angular speed, where C_D is the aerodynamic drag coefficient $d_i = \frac{1}{2}\rho\Lambda C_D r^2\omega_i^2 = d\omega_i^2$

Therefore, these forces create different torques around the pitch ϕ, the roll θ and the yaw ψ axis that are respectively given as follows, where λ is a positive coefficient [1], [7], [8], [6]). The total thrust force F_z is applied on the Quadrotor body in the z-axis.

$$\begin{aligned} F_z &= b\sum_{i=1}^{4} F_i \\ \Gamma_\phi &= l\left(F_4 - F_2\right) \\ \Gamma_\theta &= l\left(F_3 - F_1\right) \\ \Gamma_\psi &= \lambda\left(F_1 - F_2 + F_3 - F_4\right) \end{aligned} \tag{1}$$

These force and torques are provided through four Brushless DC motors which are characterized by a high torque and low friction [7]. These motors behave, at the steady state regime, like conventional DC motor. Hence, the armature voltage of the i^{th} Brushless DC motor is defined as follows:

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

where R_{mot} and k_{mot} denote the internal resistance and torque coefficient of the Brushless motors, d is the drag propellers' coefficient. Since that the drag coefficient d is very small, this dynamic can be approximated as a linear transfer function where the characteristic parameters can be identified by experimental trials as shown in ([9]).

B. Kinematics Relations

The Quadrotor is described through the body-frame $\mathbf{R}_B\left(O, x_b, y_b, z_b\right)$ and earth-frame $\mathbf{R}_E\left(o, x_e, y_e, z_e\right)$ as shown in Fig. 1 [1]. The total mass of the Quadrotor is m , g represents the gravity and l the distance from the center of each rotor to the Center of Gravity (COG).

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 $\mathbf{R}_E$. Let ν_B be the linear velocity of the Quadrotor in the body attached frame and and $\nu = (\dot{x}, \dot{y}, \dot{z})^T$ the linear velocity of the UAV in the earth-frame $\mathbf{R}_E$. The vector $\nu = (\dot{x}, \dot{y}, \dot{z})^T$ is expressed, with the rotation $\mathcal{R}$ matrix as follows:

$$\nu = \mathcal{R}\left(\phi, \theta, \psi\right).\nu_B \tag{3}$$

The rotation matrix $\mathcal{R}\left(\phi, \theta, \psi\right) : \mathbf{R}_E \rightarrow \mathbf{R}_B$ depends on the Euler angles (ϕ, θ, ψ) and is defined as follows, where $c = \cos\left(.\right)$ and $s = \sin\left(.\right)$:

$$\mathcal{R} = \begin{pmatrix} c\psi c\theta & s\phi s\theta c\psi - s\psi c\phi & c\phi s\theta c\psi + s\psi s\phi \\ s\psi c\theta & s\phi s\theta s\psi + c\psi c\theta & c\phi s\theta s\psi - s\phi c\psi \\ -s\theta & s\phi c\theta & c\phi c\theta \end{pmatrix} \tag{4}$$

It is worthwhile to note that the stability constraints give the following motions limits on the pitch $\phi \in \left[-\frac{\pi}{2}, \frac{\pi}{2}\right]$, the roll $\theta \in \left[-\frac{\pi}{2}, \frac{\pi}{2}\right]$ and the yaw $\psi \in [-\pi, \pi]$.

$\vartheta = (p, q, r)^T$ denotes the angular velocity vector 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:

$$\vartheta = \begin{pmatrix} \dot{\phi} - s\theta\dot{\psi} \\ c\phi\dot{\theta} + s\phi c\theta\dot{\psi} \\ c\phi c\theta\dot{\psi} - s\phi\dot{\theta} \end{pmatrix} \tag{5}$$

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

$$\vartheta = \begin{pmatrix} p \\ q \\ r \end{pmatrix} = \begin{pmatrix} 1 & 0 & -s\theta \\ 0 & c\phi & s\phi c\theta \\ 0 & -s\phi & c\phi c\theta \end{pmatrix} \begin{pmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{pmatrix} = T\left(\phi, \theta\right)\dot{\eta} \tag{6}$$

where T is the well known velocities' transformation matrix which is invertible under the motion limits (see after eq(4)). $\dot{\eta} = [\dot{\phi}, \dot{\theta}, \dot{\psi}]^T$

C. Dynamic Modelling

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

$$M_{gp} = \sum_{i=1}^{4} \Omega \wedge \left(0, 0, J_r \left(-1\right)^{i+1} \omega_i\right)^T \tag{7}$$

$$M_{gb} = \Omega \wedge J\Omega \tag{8}$$

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:

$$J = \begin{pmatrix} J_{xx} & 0 & 0 \\ 0 & J_{yy} & 0 \\ 0 & 0 & J_{zz} \end{pmatrix} \tag{9}$$

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

$$\begin{cases} m\ddot{\xi} = & F_{th} - F_d + F_g \\ J\dot{\Omega} = & M - M_{gp} - M_{gb} - M_a \end{cases} \tag{10}$$

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

By substituting the position vector and the forces with their expressions into Eq. (10), we have the following translations dynamics of the Quadrotor:

$$\begin{cases} \ddot{X} = \frac{1}{m}\left(c\phi c\psi s\theta + s\phi s\psi\right)u_1 - \frac{\kappa_1}{m}\dot{X} \\ \ddot{Y} = \frac{1}{m}\left(c\phi s\psi s\theta - s\phi c\psi\right)u_1 - \frac{\kappa_2}{m}\dot{Y} \\ \ddot{Z} = \frac{1}{m}c\phi c\theta u_1 - g - \frac{\kappa_3}{m}\dot{Z} \end{cases} \tag{11}$$

From the second part of Eq. (10), and while substituting each moment by its expression, we deduce the following rotational dynamics of the rotorcraft:

$$\begin{cases} \dot{p} = qr\frac{J_{yy} - J_{zz}}{J_{xx}} - \frac{J_r}{J_{xx}}\omega_r q - \frac{\kappa_4}{J_{xx}}p + \frac{1}{J_{xx}}u_2 \\ \dot{q} = pr\frac{J_{zz} - J_{xx}}{J_{yy}} + \frac{J_r}{J_{yy}}\omega_r p - \frac{\kappa_5}{J_{yy}}q + \frac{1}{J_{yy}}u_3 \\ \dot{r} = pq\frac{J_{xx} - J_{yy}}{J_{zz}} - \frac{\kappa_6}{J_{zz}}r + \frac{1}{J_{zz}}u_4 \end{cases} \tag{12}$$

According to the established equations (11) and (12), $x = \left(X, Y, Z, \phi, \theta, \psi, \dot{X}, \dot{Y}, \dot{Z}, p, q, r\right)$ is retained as the state-space vector of the nonlinear model of the Quadrotor rewritten as the following form:

$$\dot{x} = f\left(x, u\right) : \begin{cases} \dot{x}_1 = x_7 \\ \dot{x}_2 = x_8 \\ \dot{x}_3 = x_9 \\ \dot{x}_4 = x_{10} \\ \dot{x}_5 = x_{11} \\ \dot{x}_6 = x_{12} \\ \dot{x}_7 = a_9 x_7 + \frac{1}{m}u_x u_1 \\ \dot{x}_8 = a_{10} x_8 + \frac{1}{m}u_y u_1 \\ \dot{x}_9 = a_{11} x_9 + \frac{c\phi c\theta}{m}u_1 - g \\ \dot{x}_{10} = a_1 x_{11} x_{12} + a_2 x_{10} + a_3 \omega_r x_{11} + b_1 u_2 \\ \dot{x}_{11} = a_4 x_{10} x_{12} + a_5 x_{11} + a_6 \omega_r x_{10} + b_2 u_3 \\ \dot{x}_{12} = a_7 x_{10} x_{11} + a_8 x_{12} + b_3 u_4 \end{cases} \tag{13}$$

where:

$$\begin{cases} u_x = c\phi c\psi s\theta + s\phi s\psi \\ u_y = c\phi s\theta s\psi - s\phi c\psi \end{cases} \tag{14}$$

and,
$a_1 = \left(J_{yy} - J_{zz}\right)/J_{xx}$, $a_2 = -\kappa_4/J_{xx}$, $a_3 = -J_r/J_{xx}$,
$a_4 = \left(I_z - I_x\right)/J_{yy}$, $a_5 = -\kappa_5/J_{yy}$, $a_6 = -J_r/J_{yy}$,
$a_7 = \left(I_y - I_x\right)/J_{zz}$, $a_8 = -\kappa_6/J_{zz}$, $a_9 = -\kappa_1/m$,
$a_{10} = -\kappa_2/m$, $a_{11} = -\kappa_3/m$,
$b_1 = -l/J_{xx}$, $b_2 = -l/J_{yy}$, $b_3 = -l/J_{zz}$,

Note that $\kappa_{1,2,\ldots,6}$ are the aerodynamic friction and translations drag coefficients. The overall residual rotor angular speed is $\omega_r = \omega_1 - \omega_2 + \omega_3 - \omega_4$ and $x = \left(x_1, x_2, x_3, x_4, x_5, x_6, x_7, x_8, x_9, x_{10}, x_{11}, x_{12}\right)^T$ is the above Quadrotor state-vector.

The internal inputs $\left(u_x, u_y\right)$ are desidned to deduce the required $\left(\phi_d, \theta_d\right)$ which gives the UAV the good flying state posture.

D. Sensors Dynamics

The Quadrotor states are measured using an Inertial Measurement Unit (IMU) which contains accelerometers and gyroscope sensors [9], [7]. These give us measurements of the translational and rotational velocities [8].

The acceleration and velocity sensors parameters are the gains $\left(\alpha_a, \alpha_g\right)$, the bias $\left(\beta_a, \beta_g\right)$ and the additive noise assumed white and centered $\left(\gamma_a, \gamma_g\right)$, respectively.

The translation and rotation outputs acceleration a_m and gyrometer g_m measurements along x, y, and z axes can be described by Eq. (15) and Eq. (16) respectively.

$$a_m = \alpha_{acc}\nu_B + \beta^{acc} + \gamma^{acc} \tag{15}$$

$$g_m = \alpha^{gyro}\vartheta + \beta^{gyro} + \gamma^{gyro} \tag{16}$$

where $a_m = \left(a_X^m, a_Y^m, a_Z^m\right)^T$ are the sensor outputs,
$\alpha^{acc} = diag\left(\alpha_X^{acc}, \alpha_Y^{acc}, \alpha_Z^{acc}\right)$: the accelerometer gains,
$\nu_B = \mathcal{R}^{-1}\left(\phi, \theta, \psi\right)\nu$: linear velocities in the body-frame,
$\beta^{acc} = \left(\beta_X^{accAustin}, \beta_Y^{acc}, \beta_Z^{acc}\right)^T$are the sensor bias and
$\gamma^{acc} = \left(\gamma_X^{acc}, \gamma_Y^{acc}, \gamma_Z^{acc}\right)^T$ are zero mean white noises.

where $g_m = \left(g_X^m, g_Y^m, g_Z^m\right)^T$ are the sensor outputs,
$\alpha^{gyro} = diag\left(\alpha_X^{gyro}, \alpha_Y^{gyro}, \alpha_Z^{gyro}\right)$ are the gyroscope gains,
$\vartheta = \left(p, q, r\right)^T$ are 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
$\gamma^{gyro} = \left(\gamma_X^{gyro}, \gamma_Y^{gyro}, \gamma_Z^{gyro}\right)^T$ zero mean white noises.

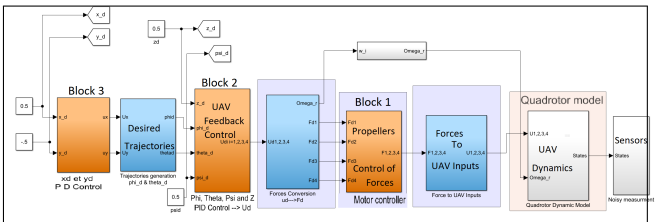

Fig. 2. Control Structure of the UAV

III. UAV Control Strategy

A. The UAV Control Structure

The UAV trajectories to be followed are, in our case generated by IBVS, applying the function control approach [10][11][6][12], [13].

The control strategy of the whole system is summarized by the Fig. 2 in three steps:
- the motors control (block 1 and variables transformations),
- the inner loop for UAV states Feedback (block 2) and
- the desired trajectories definition (block 3 and inputs selection).

In the case of target tracking the block 4 is added for trajectories generation by **IBVS**. Please note that block 1 and block 3 will be common for all the four control approaches compared in this paper. Four control design approach are compared in the block 2.

The **controllability** of the UAV dynamics needs first the selection of adequate inputs signals u_i. The chosen 4 control inputs of the Quadrotor dynamics are thus defined as follows, with $u = (u_1, u_2, u_3, u_4)^T$:

$$u = \begin{pmatrix} F \\ \Gamma_\phi \\ \Gamma_\theta \\ \Gamma_\psi \end{pmatrix} = \begin{pmatrix} b & b & b & b \\ 0 & -lb & 0 & lb \\ -lb & 0 & lb & 0 \\ d & -d & d & -d \end{pmatrix} \begin{pmatrix} \omega_1^2 \\ \omega_2^2 \\ \omega_3^2 \\ \omega_4^2 \end{pmatrix} \tag{17}$$

The inputs u_2 , u_3 and u_4 represent the controls torques of the pitch, the roll and yaw, respectively. Please note that there are 4 inputs and 6 different position outputs.

Then the basic level of control is to produce these inputs by controlling each one of the 4 motors to produce the propellers forces F_i by means of velocities ω_i and then generate inputs (the torques and forces u_i of equation (2)). This is because we have chosen to control, as main outputs the variables (z, ϕ, θ, ψ). Please not that we can select 4 outputs main state variables to be controlled and then two remaining states will be controlled in a second step (block 3) of control as internal states.

Unlike the commonly used angular speed control of the motors, here, we choose the propelling forces F_i as actuators outputs to be controlled, by means of the motors voltage inputs v_i. The thrust forces F_i must follow the desired forces F_i^d. This is the first control step, which is common to all our control approaches. The motors and propellers are consider as force actuators and then a PD control ensures that the output forces F_i follow the desired ones F_i^d as it shown in Fig 3 (see also the block 1 for motor control in fig 4. This

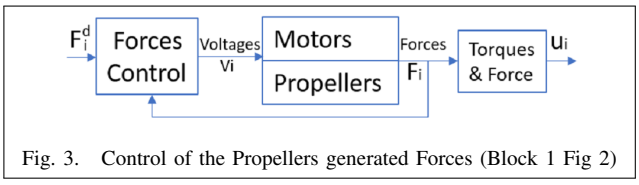

Fig. 3. Control of the Propellers generated Forces (Block 1 Fig 2)

has allowed to get better results than the literature reported ones (faster responses and convergence) [14], [1].

As the UAV has four inputs u_i and six output variables to be controlled, the decoupling strategies and feedback linearization, which require squared systems, need selection of 4 outputs to be decoupled. Generally one use the x, y, z positions and ψ the steering angle to track heading and desired positions and orientation. The states z and ψ are controlled directly by feedback in block 2.

The remaining two angles ϕ and θ are considered as internal system states. They cannot be controlled arbitrarily and their reference signals have to be deduced from the required outputs in x, y directions, respectively. Then, ϕ and θ require an inner control loop which is the second control step, see the first and second block in Fig (2). The desired ϕ_d and θ_d are deduced by means of a control on x and y in order to follow x_d and y_d (2)).

B. PID Feedback Control

The PID is extensively used in a variety of aerial applications relating to the piloting of UAV drones [15], [16]. We can find in litterature, the design of a PID stabilizing control for the attitude control (ϕ, θ, ψ) of an UAV around a fixed point in space. Other application of PID regulation has been proposed to stabilize the flight of a drone in space according to the angles of roll and pitch only. The authors often use a linearized model for stabilizing control based on a desired attitude and position. Some authors developed a technique of cascading the two PID regulators to control the x and y coordinates in order to control the attitude of a drone.

In our case PID control laws are used in the three blocks 1, 2 and 3 of Fig (2). A parallel architecture of PIDs has been used. The controller structure is made of two cascaded control loops (and the motors control). In the inner control loop, the attitude and the altitude are driven using a PID controller, whereas the positions x and y are efficiently controlled by applying a PD control. The roll and pitch PID controllers read the commands from the PD controller outputs u_x and u_y, respectively. During the simulation, we have tested two cases; when u_x and u_y are directly used as desired trajectories and by computing the equivalent control using the relationship between these commands and angles according to Eq.14. The last case leads to less oscillations in steady-state phase. The deduced control expressions are as follows:

$$\begin{aligned} u_1 &= k_{p_z}(z - z_d) + k_{i_z} \int (z - z_d) + k_{d_z}(\dot{z} - \dot{z_d}) \\ u_2 &= k_{p_\phi}(\phi - \phi_d) + k_{i_\phi} \int (\phi - \phi_d) + k_{d_\phi}(\dot{\phi} - \dot{\phi_d}) \\ u_3 &= k_{p_\theta}(\theta - \theta_d) + k_{i_\theta} \int (\theta - \theta_d) + k_{d_\theta}(\dot{\theta} - \dot{\theta_d}) \end{aligned}$$

$$\begin{aligned} u_4 &= k_{p_\psi}(\psi-\psi_d)+k_{i_\psi}\int(\psi-\psi_d)+k_{d_\psi}(\dot{\psi}-\dot{\psi_d}) \\ u_x &= k_{p_x}(x-x_d)+k_{d_x}(\dot{x}-\dot{x_d}) \\ u_y &= k_{p_y}(y-y_d)+k_{d_y}(\dot{y}-\dot{y_d}) \end{aligned} \quad (18)$$

C. Feedback Linearization

Feedback linearization will transform the nonlinear system into linear and non interacting subsystems using the Lie derivatives to get the Brunovsky form, where $v^T = (v_1, v_2, v_3, v_4)$, are the linearized system control inputs

$$\begin{array}{l} \frac{d^4}{dt^4}x = v_1 \\ \frac{d^4}{dt^4}y = v_2 \\ \frac{d^4}{dt^4}z = v_3 \\ \frac{d^2}{dt^2}\psi = v_4 \end{array} \quad (19)$$

From equation (19), the linearized control laws computed after four derivatives steps makes this controller highly sensitive to disturbance. The motor and sensors dynamics are often discarded, in literature, and only the UAV model is used during the simulation is used to compute the control. It is proved in [14] that the UAV system given by Eq.(13) is not linearizable by static state feedback. To overcome this problem, the system dynamics has been extended by including two internal variables as follows:

$$\left\{ \begin{array}{rcl} u_1 &=& \zeta \\ \dot{\zeta} &=& \xi \\ \dot{\xi} &=& \overline{u}_1 \end{array} \right. \quad (20)$$

where u_1 is not anymore an input but becomes the internal state variable ζ. The extended system is described by:

$$\dot{\underline{x}} = \overline{f(\underline{x})} + \sum_{i=1}^{4} \overline{g_i(\underline{x})}\overline{u}_i \quad (21)$$

where $\underline{x} = \Big(X, Y, Z, \phi, \theta, \psi, \dot{X}, \dot{Y}, \dot{Z}, p, q, r, \zeta, \xi\Big)$,
$\overline{f(\underline{x})} = [f(x), \xi, 0]^T$, $\overline{g}_1 = [0, ..., 0, 1]^T$,
$\overline{g}_2 = [0, ..., b_1, 0, 0, 0]^T$, $\overline{g}_3 = [0, ..., b_2, 0, 0]^T$,
$\overline{g}_4 = [0, ..., b_3, 0]^T$ and $\overline{u}_{2,3,4} = u_{2,3,4}$.
We have: $[x^{(4)}, y^{(4)}, z^{(4)}, \psi^{(2)}] = \mathbf{b}(\underline{x}) + \mathbf{a}(\underline{x})\overline{u}$ where $\mathbf{a}(\underline{x})$ and $\mathbf{b}(\underline{x})$ are defined using the Lie derivatives.

The dynamic state feedback control used is of the form:

$$\overline{u} = -\mathbf{a}(\underline{x})^{-1}\mathbf{b}(\underline{x}) + \mathbf{a}(\underline{x})^{-1}v \quad (22)$$

D. Back-Stepping Control

Backstepping approach is a recursive control that consists to create (step by step) intermediate variables for a group of selected states that are called virtual controls of the studied system [3], [17]. Among the works that implement the Backstepping on the real systems we have found in the literature [18], [19]. The authors performed an online optimization based on Backstepping control to solve the problem of virtual controls at each step.

The work in [3] illustrates the position control of the drone. In order to verify the effectiveness of the proposed back-stepping controller, regard to the previously proposed method, we introduce uncertainties in the model parameters and measurement noise.

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

The whole system is asymptotically stable with the following control laws:
$u_1 = \frac{m}{\cos(\overline{x}_4)\cos(\overline{x}_5)}(g - \overline{a}_{11}\overline{x}_9 + \ddot{z}_d + k_{11}(-k_{11}e_{11} + e_{12}) + k_{12}e_{12} + e_{11})$
$u_2 = \frac{1}{\overline{b}_1}(-\overline{a}_1\overline{x}_{11}x_{12} - \overline{a}_2\overline{x}_{10} - \overline{a}_3\omega_r\overline{x}_{11} + \ddot{\phi}_d + k_1(-k_1e_1 + e_2) + k_2e_2 + e_1)$
$u_3 = \frac{1}{\overline{b}_2}(-\overline{a}_4\overline{x}_{10}\overline{x}_{12} - \overline{a}_5\overline{x}_{11} - \overline{a}_6\omega_r\overline{x}_{10} + \ddot{\theta}_d + k_3(-k_3e_3 + e_4) + k_4e_4 + e_3)$
$u_4 = \frac{1}{\overline{b}_3}(-a_7\overline{x}_{10}\overline{x}_{11} - \overline{a}_8\overline{x}_{12} - \overline{a}_3\omega_r\overline{x}_{11} + \ddot{\psi}_d + k_5(-k_5e_5 + e_6) + k_6e_6 + e_5)$
$u_x = \frac{m}{u_1}(-\overline{a}_9\overline{x}_7 + \ddot{x}_d + k_7(-k_7e_7 + e_8) + k_8e_8 + e_7)$
$u_y = \frac{m}{u_1}(-\overline{a}_{10}\overline{x}_8 + \ddot{y}_d + k_9(-k_9e_9 + e_{10}) + k_{10}e_{10} + e_9)$
Where $e_i = \overline{x}_i - x_{i_d}$ and $\overline{x}_i$ is the state variable subject to bias and noise measurement.

E. SMC Control

The Sliding Mode Control is one of robust approach that produces a switching control law in order to drive the state trajectories and maintain the system on a defined sliding manifold [20], [21].

Obviously the state vector need the primary state (x_0, y_0, z_0, ψ) and their successive derivatives. On adopting a classical polynomial based control law, the closed loop system will be represented by the following equations:
$u_1 = \frac{m}{\cos(\overline{x}_4)\cos(\overline{x}_5)}(g + \dot{x}_{9_d} - \overline{a}_{11}\overline{x}_9 - \lambda_z(\overline{x}_9 - x_{9_d})) - k_z sign(S_z)$
$u_2 = \frac{1}{\overline{b}_1}(\dot{x}_{10_d} - \overline{a}_1\overline{x}_{11}\overline{x}_{12} - \overline{a}_2\overline{x}_{10} - \overline{a}_3\omega_r\overline{x}_{10} - \lambda_\phi(\overline{x}_{10} - x_{10_d})) - k_\phi sign(S_\phi)$
$u_3 = \frac{1}{\overline{b}_2}(\dot{x}_{11_d} - \overline{a}_4\overline{x}_{10}\overline{x}_{12} - \overline{a}_5\overline{x}_{11} - \overline{a}_6\omega_r\overline{x}_{10} - \lambda_\theta(\overline{x}_{11} - x_{11_d})) - k_\theta sign(S_\theta)$
$u_4 = \frac{1}{\overline{b}_3}(\dot{x}_{12_d} - \overline{a}_7\overline{x}_{10}\overline{x}_{11} - \overline{a}_{10}\overline{x}_{12} - \lambda_\psi(\overline{x}_{12} - x_{12_d})) - k_\psi sign(S_\psi)$
$u_x = \frac{m}{u_1}(\dot{x}_{7_d} - \overline{a}_9\overline{x}_7 - \lambda_x) - k_x sign(S_x)$
$u_y = \frac{m}{u_1}(\dot{x}_{8_d} - \overline{a}_{10}\overline{x}_8 - \lambda_y) - k_y sign(S_y)$
where $S_i = \dot{e}_i + \lambda e_i$ with $e_i = \overline{x}_i - x_i$.

IV. Simulation Results

All the following simulations are run on the same PC with 2.2.GH Intel processor and 8GB RAM using Simulink tools of Matlab. The desired trajectories and simulation parameters are also unified. Only three positions x, y, z and the yaw angle ψ are depicted for all control strategies. Excepting feedback linearization control, the simulation results of remaining control strategies show three responses; the sensor output signal used in the process control, the model output referring to the ideal case without perturbation measurement and the desired trajectories.
The Quadrotor and motors parameters used in all our simulation are reported in Table.I. 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

Parameter	Value	Parameter	Value
m	0.486	g	9.81
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 I

QUADROTOR PARAMETERS VALUES

block generator with noise power equal to 10^{-4}. The desired trajectories are chosen to not to be important in the aim to see clearly the effect of the sensor dynamic on the responses and consequently on the control robustness. Maintaining the same simulation parameters, the integration step is fixed to 1 ms with the same differential solver.

A. PID control

The simulation results for PID control structure beforehand proposed are depicted in Fig. 4. Despite the finite time of the transition response, it is not possible to improve this time due to the saturation applied at the desired trajectories for pitch and roll angles. The PID parameters for each variable are reported in Table.II.

PID coefficients	ϕ	θ	ψ	x	y	z
k_p	5.5	5.5	0.1	1.2	1.2	25
k_i	0.5	0.5	0.1	X	X	15
k_d	1	1	0.1	1	1	5

TABLE II

PIDS' PARAMETER VALUES

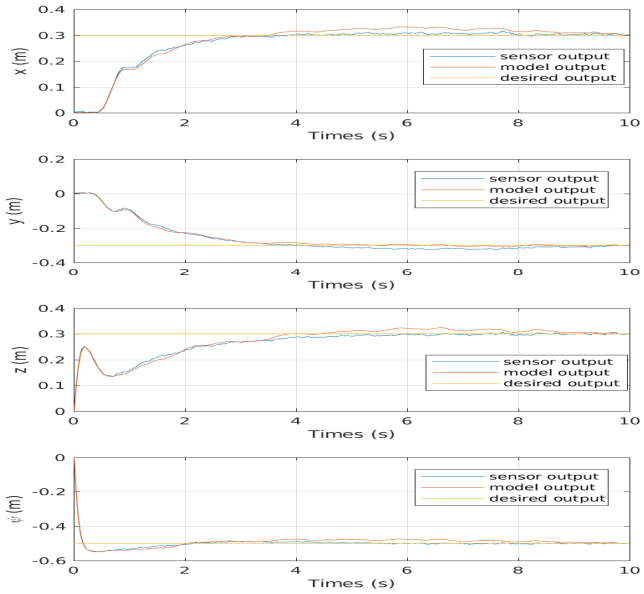

Fig. 4. x,y, z and ψ tracking trajectories using PID controller

B. Feedback Linearization Control (FLC)

One way to design a linear tracking controller is by placing the poles properly to satisfy stability. That is,

$P_1 = P_2 = P_3 = (s + p_1)^4$ and $P_4 = (s + p_2)^2$.
Figure 5 shows the response behavior for different values of p_1. Despite the stabilizing control laws $v_{1,2,3}$ are selected to evolve as a first order system, the responses show a second order system behavior, even if p_1 is too small. Changing the pole p_1 affects only steady-state time response and maintains the second order behavior. The yaw angle is not affected by the choice of p_1. The corresponding pole is selected as $p_2 = 5$. The execution time of this control is quite long. Then it cannot be used in real application.

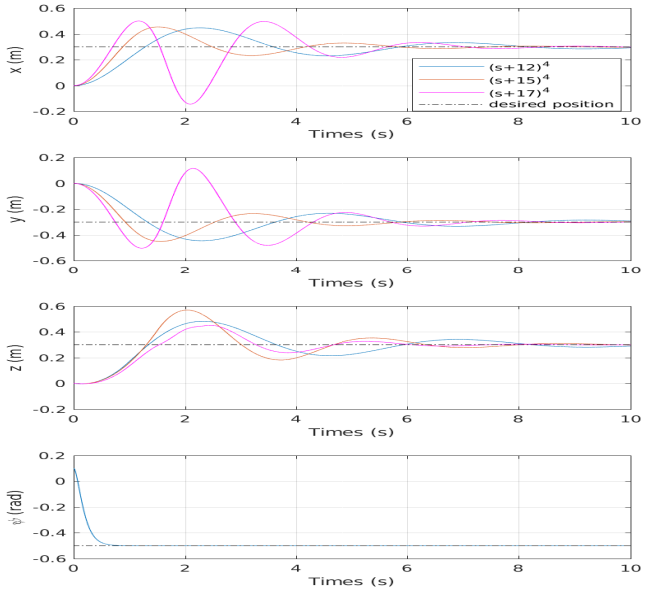

Fig. 5. x,y,z and ψ tracking trajectories using input-output feedback linearization controller

C. Back-Stepping Control (BSC)

The following backstepping controller parameters are used in simulation: $k_1 = k_2 = k_3 = k_4 = 15$, $k_5 = 10$, $k_6 = 2$, $k_7 = k_9 = 5$, $k_8 = k_{10} = 1$, $k_{11} = 10$ and $k_{12} = 5$. Unlike the PID controller, the backstepping is able to drive a large variation scale of roll and pitch angles. This leads to shorter steady state response than PID controller. Moreover, backstepping controller shows smooth responses and less oscillations around the desired trajectories (see Fig.6).

D. Sliding Modes Control (SMC)

This subsection shows the SMC robustness against some nonlinear phenomena considered during the simulation of the whole system and neglected in control design. Once again, the simulation conditions are similar to the ones applied for BSC and PID controllers. The controller parameters are presented in Table (III). Figure 7 depicts the simulation of

SMC coefficients	ϕ	θ	ψ	x	y	z
k	0.1	0.1	3	0.1	0.1	3
λ	1.5	1.5	3.5	0.5	0.5	3

TABLE III

SMC PARAMETERS VALUES

the SMC controller. The obtained results show less effects

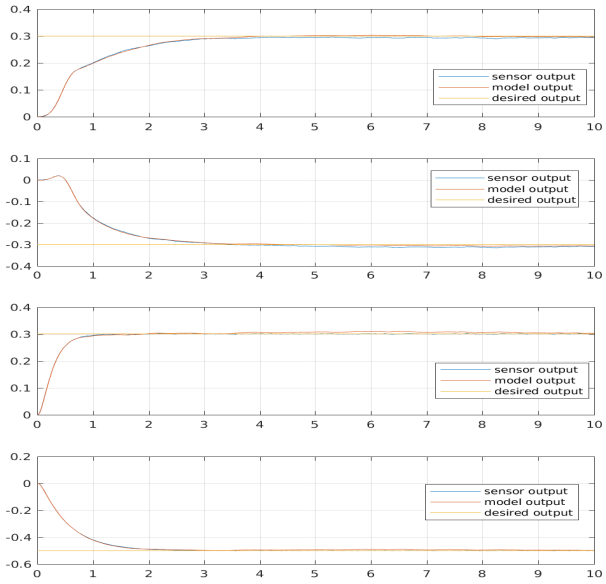

Fig. 6. x,y, z and ψ tracking trajectories using back-stepping control

against the perturbation and chattering despite the steady state reaching time is relatively large

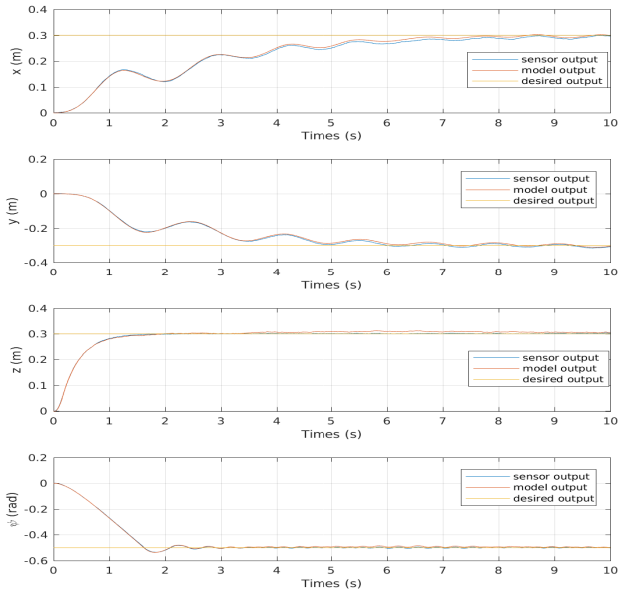

Fig. 7. x,y, z and ψ tracking trajectories using SMC

E. Comparative Analysis

The controllers PID, BSC and SMC are designed to control the Quadrotor under the same operation conditions. Criteria to select one of these approaches depend on the goals and the operating constraints such as power consumption or computation burden (processing time in Matlab). The simulation of the UAV real time behaviour is done in Maltab/Simulink, as presented in section 2 fr all the tests.

An important factor to be compared is the power consumption which can be computed from the forces F_i: $P \simeq \int_0^{10}(|F_1|+|F_2|+|F_3|+|F_4|)dt$. Other factors can also taken into account such as: sensitivity against noise, implementation difficulty, environment nonlinearity robustness which are sumarized in Table.IV.

Criteria	PID	F.L.C	B.S.C	SMC
Simulation run Time	○	●	●	●
Power consumption	●	●	●	○
Stabilization time	○	●	●	●
Sensor dynamics effect	●	●	○	●
Implementation difficulty	○	●	●	●

TABLE IV
COMPARATIVE STUDY

They are classified from the best (in white) to the worst(in red) result: ○, ●, ●, ●.

The average degree of the preference indicates that the PID and SMC have the are preferred. The second rank is the BSC and the last one is the FLC that remains a theoretical approach difficult to implement.

V. VISUAL TRACKING COMPARATIVE STUDY

The most important Quadrotor task is the tracking of (static or mobile) objects using the visual sensors. The visual control is a very good alternative in the presence of drifts and measurement errors. Here, IBVS is used to control the UAV. Let the UAV position considered relatively to the object and control the motion with regard to this distance. We use the visual projection in pixels:

$$e(t) = s(t) - s(t)^* \tag{23}$$

The control task function (minimization criterion in function of $e(t)$) is then defined in image plane as the error between a desired and actual relative position of the target (object) ([10], [11], [22]).

The feedback law leading the linear and angular velocities $V = [v_x, v_y, v_z, \omega_x, \omega_y, \omega_z]$ needed to track the object is defined as follows:

$$V = -\sigma \mathbf{L}^+ e(t) \tag{24}$$

where $\mathbf{L}^+$ is the pseudo-inverse of the interaction matrix and σ is a convergence rate factor (see [6]). V is then used to define the UAV desired trajectories (to be followed by UAV the controllers).

Our aim is to evaluate the controllers when performing a visual tracking of mobile object which is featured by four points in the image of a virtual camera. The desired position for the UAV is chosen to track the center of the object at 1 m altitude.

The object follow a circular trajectory and we assuming that the frame of the virtual camera is the Quadrotor one.

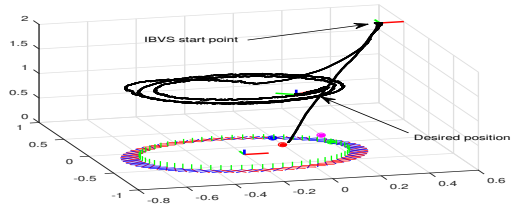

Fig. 8. Visual Tracking for circular trajectory using PID controller

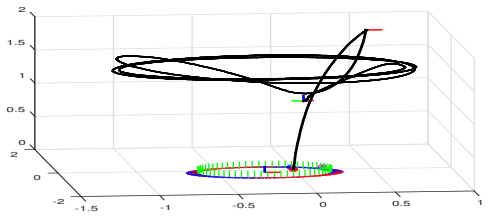

Fig. 9. Visual Tracking for circular trajectory using Back-Stepping control

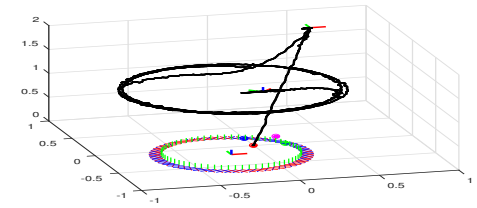

Fig. 10. Visual Tracking for circular trajectory using SMC controller

The figure 8 shows that the PID control leads to trajectories as near as possible to the circular motion of the tracked object. Nevertheless, these trajectories oscillate more than BSC and than SMC. An advantage of the PID control is that the altitude of the trajectories during the transition is near to the desired one, whereas, the BSC and SMC have an altitude relatively far from the desired one. The PID control leads better performances in visual tracking tasks. The feedback linearization has not been tested due to its computing cost.

VI. CONCLUSIONS

In this paper, we have compared the performances of PID, FLC, BSC and SMC for Quadrotor control. A suitable and efficient structure is proposed for implementation of the control laws. Aerodynamics effects that can influence the evolution of the Quadrotor have been taken into account, as well as different components such as the actuators dynamics, their control (force control) and the sensors dynamics and perturbations. The comparative study shows that, with our control structure, PID and SMC give better results rather than the and FLC. Whereas, the LFC needs a heavy computing cost. Use of Image-Based Visual Tracking (IBVT) for a mobile target can compensate the sensors drifts and errors. The need of observers is then avoided. In this Comparative Analysis, the PID allows to follow the object with less errors compared to SMC and BSC. With the proposed structure for control, the PID control may be enough. Thanks to the control structure proposed which optimizes the controlability of the system (see Fig (2)).

ACKNOWLEDGMENT

This work is supported by the SASV 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.

REFERENCES

[1] V. Mistler, A. Benallegue, and N. 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*, Bordeaux and Paris, 2001, pp. 586–593.

[2] V. Mistler, A. Benallegue, and N. M'Sirdi, "Linearisation exacte et decouplage entrees-sorties comparaison entre l'helicoptere standard et l'helicoptre 4 rotors." in *Proceedings of the CIFA 2002*, Nantes, 2002.

[3] L. Mederreg, F. Diaz, and N. M'Sirdi, "Nonlinear backstepping control with observer design for 4 rotors helicopter." in *Proceedings of AVCS 2004*, Genova, 2004. [Online]. Available: http://nkms.free.fr/.NkMs/IAV04-Paper4Rotors.pdf

[4] L. Mederreg, F. Diaz, and N. M'Sirdi, "Dynamic feedback control for a quadrotor unmanned aerial vehicle," in *SSD 2005*, Sousse, Tunisia, 2005.

[5] A. Mokhtari, N. K. M'Sirdi, K. Meghriche, and A. Belaidi, "Feedback linearization and linear observer for a quadrotor unmanned aerial vehicle," *Advanced Robotics*, vol. 20, no. 1, pp. 71–91, 2006. [Online]. Available: https://doi.org/10.1163/156855306775275495

[6] C. Bensalah, N. M'Sirdi, and A. Naamane, "Full modelling and sliding mode control for a quadrotor uav in visual servoing task," in *IMAACA 2019*, Lisboa, 2019.

[7] R. Austin, *Unmanned Aircraft Systems: UAVs design, development and deployment*, J. W. 1st edition and S. Ltd, Eds. John Wiley and Sons Ltd, London, 2010.

[8] A. Guerrero and R. Lozano, *Flight formation control*, edition of John Wiley and S. Ltd, Eds. John Wiley and Sons Ltd, London, 2012.

[9] M. Becker, R. Coronel B. Sampaio, V. Bouabdallah, S.and 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*, vol. 34, no. 3, pp. 294–307, 2012.

[10] C. Samson and B. Espiau, "Application of the task-function approach to sensor-based control of robot manipulators," *IFAC Proceedings Volumes*, vol. 23, no. 8, Part 5, pp. 269 – 274, 1990, 11th IFAC World Congress on Automatic Control, Tallinn, 1990 - Volume 5, Tallinn, Finland. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1474667017517462

[11] F. Chaumette and S. Hutchinson, "Visual servo control part i: basic approaches," *IEEE Robot. Autom. Mag*, vol. 13, no. 4, 2006.

[12] T. Hamel and R. Mahony, "Visual servoing of an under-actuated dynamic rigid-body system: an image-based approach," *IEEE Trans. Robot. Autom*, vol. 18, no. 2, 2006.

[13] S. Azrad, F. Kendoul, and K. Nonami, "Visual servoing of quadrotor micro-air vehicle using color-based tracking algorithm," *J. Syst. Des. Dyn.*, vol. 4, no. 2, 2010.

[14] A. Mokhtari, N. M'Sirdi, K. Meghriche, and A. Belaidi, "Feedback linearization and linear observer for a quadrotor unmanned aerial vehicle," *Advanced Robotics*, vol. 20, no. 1, 2006.

[15] J. Li and Y. Li, "Dynamic analysis and pid control for a quadrotor," in *Proc. of IEEE International Conference on Mechatronics and Automation*, Beijing, China, 09 2011, p. 573578.

[16] Z. A. A. and H. D., "Simple approach on implementing imu sensor fusion in pid controller for stabilizing quadrotor flight control," in *EEE 7th International Colloquium on Signal Processing and its Applications*, Penang, Malaysia, 03 2011.

[17] F. Gavilan, J. A. Acosta, and R. Vazquez, "Control of the longitudinal flight dynamics of an uav using adaptive backstepping," in *Preprints of the 18th IFAC World Congress*, Milano (Italy), August 2011, pp. 1891–1897.

[18] M. Huang, B. Xian, C. Diao, K. Yang, and Y. Feng, "Adaptive tracking control of underactuated quadrotor unmanned aerial vehicles via backstepping," in *American Control Conference*, Baltimore, USA, 06 2010, p. 20762081.

[19] 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.

[20] N. M'Sirdi and N. Nadjar-Gauthier, *Application of Sliding Mode Control to Robotic Systems*, ser. Control Engineering Series. New York: Marcel Dekker, 2002, ch. 13, pp. 351–387.

[21] S. Emelyanov, "Theory of variable-structure control systems: inception and initial development," *Computational Mathematics and Modeling*, vol. 18, no. 4, 2007.

[22] S. Hutchinson, G. Hager, and P. Corke, "A tutorial on visual servo control," *IEEE TRANSACTIONS ON ROBOTICS AND AUTOMATION*, vol. 12, no. 5, 1996.