Vision-based Sliding Mode Control with Exponential Reaching Law for Uncooperative Ground Target Searching and Tracking by Quadcopter

Hamza Bouzerzour[1][a], Mohamed Guiatni[1][b], Mustapha Hamerlain[2] and Ahmed Allam[3][c]

[1]*Complex Systems Control and Simulators Laboratory, Ecole Militaire Polytechnique, Algiers, Algeria*

[2]*Centre de Développement des Technologies Avancées, Algiers, Algeria*

[3]*Control Process Laboratory, National Polytechnic School, Algiers, Algeria*

Keywords: Quadcopter, Exponential Reaching Law, IBVS, Sliding Mode Control, Ground Target Searching and Tracking.

Abstract: This paper propose a robust approach based on vision and sliding mode controller for searching and tracking an uncooperative and unidentified mobile ground target using a quadcopter UAV (QUAV). The proposed strategy is an Image-Based Visual Servoing (IBVS) approach using target's visual data projected in a virtual camera combined with the information provided by the QUAV's internal sensors. For an effective visual target searching, a circular search trajectory is followed, with a high altitude using the Camera Coverage Area (CCA). A Sliding Mode Controller (SMC) based on Exponential Reaching Law (ERL) is used to ensure the QUAV control in the presence of external disturbances and measurement uncertainties. Simulation results are presented to assess the proposer strategy considering different scenarios.

1 INTRODUCTION

Recently, numerous researchers have been interested in Quadrotor Unmanned Aerial Vehicles (QUAVs) challenges. The QUAVs are simpler and offer several advantages in aspects of maneuverability and flight stability. They are consequently, used in diverse applications, such as transportation (Menouar et al., 2017), real-time monitoring (Duggal et al., 2016)(Radiansyah et al., 2017), military reconnaissance (Samad et al., 2007) and inspection (Wang et al., 2022). Mobile Target Tracking is one of the applications that attract enormous attention, since it is used in multiple areas, such as, rescue operations (Cantelli et al., 2013), search and track individuals/vehicles (Prabhakaran and Sharma, 2021)(Puri, 2005) and aerial convoys tracking (Ding et al., 2010).

For target tracking, most researchers have mainly focused on tracking a cooperative target, where its trajectory is available. Nonetheless, for the uncooperative target it is still a challenging issue. When the target is occurring suddenly and its information such as the trajectory and geometrical information is not accessible, the only available solution to detect and track it, is the use of the information provided by an embedded camera. This approach is known as visual Servoing (VS), which has been the subject of intensive research since the late 1980s, and it is classified into two major methods (Janabi-Sharifi and Marey, 2010). Position-Based Visual Servoing (PBVS) method, in which, the 3D Cartesian position of the target is reconstructed from image data and used to compute the control law. Generally, PBVS involves more prior knowledge of the camera calibration parameters and target geometry.
The second method is Image-Based Visual Servoing (IBVS), where the control law is computed directly in the 2D image plane. IBVS is widely used due to it being computation-friendly and to its robustness.

Many researchers have made efforts to address the IBVS for target tracking (Borshchova and O'Young, 2016)(Pestana et al., 2014). However, the proposed techniques are based on the features Jacobian (Chaumette and Hutchinson, 2006) under the assumption that the UAV and target performed a smooth movement variation. But, the use of the features Jacobian may lead to system instability since the QUAV is an agile system. The solution to overcome this issue is the use of the virtual camera (Fink et al., 2015), where the image features are projected into a virtual camera which inherits only the translation and yaw motion of the real camera.

[a] https://orcid.org/0000-0002-2218-5777
[b] https://orcid.org/0000-0002-5899-6862
[c] https://orcid.org/0000-0002-3648-9288

Bouzerzour, H., Guiatni, M., Hamerlain, M. and Allam, A.
Vision-based Sliding Mode Control with Exponential Reaching Law for Uncooperative Ground Target Searching and Tracking by Quadcopter.
DOI: 10.5220/0011353500003271
In *Proceedings of the 19th International Conference on Informatics in Control, Automation and Robotics (ICINCO 2022)*, pages 555-564
ISBN: 978-989-758-585-2; ISSN: 2184-2809

Besides the use of the virtual camera, the vision-based target tracking requires more specific controllers, since the common control algorithms are not always applicable. Therefore, nowadays many works are conducted to develop a robust controller to ensure efficient vision-based target tracking. In (Zhang et al., 2021), the image moments are defined in the virtual camera plane and used in a nonlinear model predictive control-based IBVS to track a ground target by a quadcopter. In (Cao et al., 2017), an IBVS-based Backstepping controller combined with a nonlinear trajectory observer is designed to stabilize a quadcopter above a moving non-cooperative target. Most of the works have ignored the target searching phase and have focused only on the development of controllers for the tracking phase, considering that the target is already detected.

This work proposes a new vision based approach to control the QUAV for searching and tracking of an uncooperative and unidentified mobile ground target. The main contribution is the use of the target's visual features expressed in a virtual camera. These features are combined with the QUAV's inertial measurements, camera's model, as well as the rough information of the eventual targets (shape, dimension, maximum velocity,...etc.). In order to control the QUAV and to ensure robust autonomous target tracking, a Sliding Mode Controller (SMC) based on Exponential Reaching Law (ERL) is designed .

The rest of the paper is organized in seven sections. Section II presents the problem formulation. Section III presents QUAV's system description, modeling and control. Section IV deals with the visual system modeling while section V focuses on the target position and velocity estimation using visual measurements. In section VI, simulation results are given and discussed.

2 PROBLEM STATEMENT

2.1 Problem Definition

Tracking a ground target by a quadcopter requires the determination of the quadcopter's trajectory while the target remains in its visual field. For more accurate tracking, the horizontal instantaneous QUAV's position should coincide with the target's position while adjusting the QUAV's altitude to ensure that the target remains in its FOV.

Therefore, the problem is solved by minimizing the horizontal position/velocity errors between the quadcopter and the target. For an uncooperative target-tracking, it is necessary to establish a flight strategy to find the target, once the target is found, a vision-based process should be established to estimate the horizontal position/velocity errors between the target and the QUAV for which they will be used by the controller. Hence an adequate controller must be implemented to handle the proposed flight strategy.

2.2 Proposed Strategy Overview

The proposed flight strategy is depicted in Fig. 1, where the QUAV has at first to reach the area where the target is supposed to navigate. During this first phase called cross-phase, the QUAV flies at a low altitude to reduce energy consumption and avoid the wind. Once the QUAV reaches the searching area, a searching phase is automatically started. For an effective visual target searching, a circular trajectory is chosen, with a maximum quadcopter's altitude. The said circular trajectory is centered at the start point and its radius is determined according to the Camera Coverage Area (CCA). Once the target is detected, a vision-based tracking process is automatically triggered. During the third phase called the tracking phase, the QUAV follows the target while tuning its altitude automatically for optimal target observation.

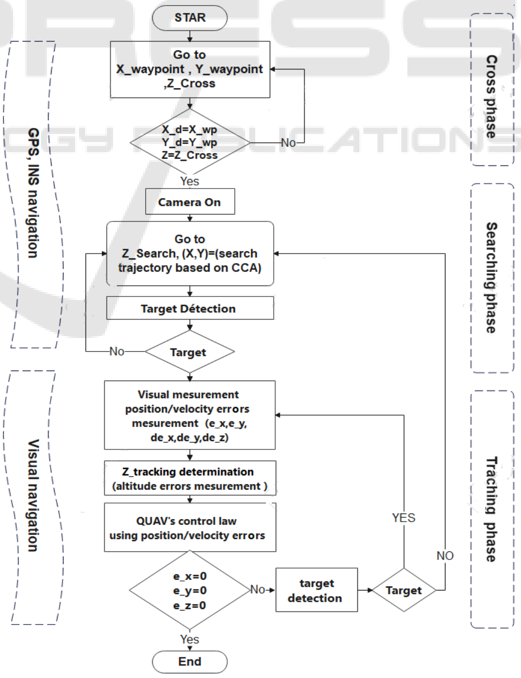
STAR
Go to X_waypoint , Y_waypoint ,Z_Cross
X_d=X_wp Y_d=Y_wp Z=Z_Cross
No
Yes
Camera On
Go to Z_Search, (X,Y)=(search trajectory based on CCA)
Target Détection
Target
No
Visual mesurement position/velocity errors mesurement (e_x,e_y, de_x,de_y,de_z)
Z_tracking determination (altitude errors mesurement)
QUAV's control law using position/velocity errors
e_x=0 e_y=0 e_z=0
No
target detection
Target
YES
NO
Yes
End
GPS, INS navigation
Visual navigation
Cross phase
Searching phase
Traching phase

Figure 1: Flowchart of the proposed strategy.

For the controller, a Sliding Mode Controller (SMC) based on Exponential Reaching Law (ERL) is selected. For the first and the second phase, the control loop uses as a reference the waypoint and the circular trajectory respectively, which will be combined with the instantaneous position and altitude of the QUAV measured by GPS, while for the tracking phase, the visual measurements are combined with Inertial measurement unit (IMU) and the altimeter sensor measurements.

3 QUADCOPTER MODELING AND CONTROL

3.1 Quadcopter Modeling

Modeling the concept of visual tracking by a QUAV, involves the definition of several frames (Fig. 2). Namely, an inertial fixed frame $F_I = (O_i, \vec{e}^{\,i}_x, \vec{e}^{\,i}_y, \vec{e}^{\,i}_z)$, a body-fixed frame $F_B = (O_b, \vec{e}^{\,b}_x, \vec{e}^{\,b}_y, \vec{e}^{\,b}_z)$ attached to the QUAV mass center, as well as a camera frame $F_C = (O_c, \vec{e}^{\,c}_x, \vec{e}^{\,c}_y, \vec{e}^{\,c}_z)$ attached to the camera's optical center.

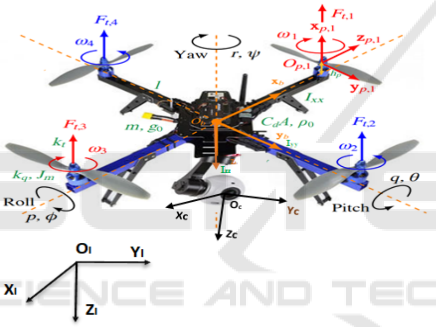

Figure 2: Quadcopter modeling and frame definition.

The quadcopter consistsof a rigid cross-frame equipped with four rotors (Fig. 2) and it is assumed symmetric with respect to the $x/y-axis$, so that the center of gravity is located at the center of the structure, where a monocular camera is fixed.

The QUAV is propelled by four forces $F_i \, (i \in \{1,2,3,4\})$ generated by the rotation of the blades mounted on the four rotors.

$$F_i = \frac{1}{2}\rho s C_T r^2 = b\omega_i^2 \tag{1}$$

With ρ and C_T are the air density and the aerodynamic thrust coefficients. s/r are the section/radius of the propeller.
The actuator rotation generates also four drag torques $\delta_i \, (i \in \{1,2,3,4\})$ which are opposed to the motor torques

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

C_T is the aerodynamic thrust coefficients.

The QUAV's attitude is controlled by three torques U_2, U_3, U_4 and the altitude is controlled by the sum of the four forces U_1 such that :

$$\begin{bmatrix} U_1 \\ U_2 \\ U_3 \\ U_4 \end{bmatrix} = \begin{bmatrix} b & b & b & b \\ -lb & 0 & lb & 0 \\ 0 & -lb & 0 & lb \\ d & -d & d & -d \end{bmatrix} \begin{bmatrix} \omega_1^2 \\ \omega_2^2 \\ \omega_3^2 \\ \omega_4^2 \end{bmatrix} \tag{3}$$

l is the distance between QUAV's gravity center and the rotor.
For the kinematic and dynamic modeling, the QUAV is considered as $6-DOF$ rigid body with mass m and a constant symmetric inertial matrix $J = diag(I_{xx}, I_{yy}, I_{zz})$. Its linear velocity in the F_I frame is denoted by $V_I = \dot{\xi} = (\dot{x}, \dot{y}, \dot{z})^t$ and is expressed in the F_b frame by the following equation:

$$v_b = R^t V_I \tag{4}$$

With $v_b = \begin{pmatrix} u & v & w \end{pmatrix}^t$. $R = R_\psi R_\theta R_\phi$ is the rotation matrix between the F_b and F_I frames.

The relation between the derivative of the Euler angles and the QUAV's angular velocity $\Omega = \begin{pmatrix} p & q & r \end{pmatrix}^t$ is given by the following equation:

$$\begin{bmatrix} p \\ q \\ r \end{bmatrix} = \begin{bmatrix} 1 & 0 & -S_\theta \\ 0 & C_\phi & S_\phi C_\theta \\ 0 & -S_\phi & C_\phi C_\theta \end{bmatrix} \begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} \tag{5}$$

Using the Euler Newton formalism, the kinematic equation is given by:

$$\begin{cases} m\ddot{\xi} = F_t - F_d + F_g \\ J\dot{\Omega} = M_f - M_{gp} - M_{gb} - M_a \end{cases} \tag{6}$$

- $F_t = R.\begin{bmatrix} 0 & 0 & -b\left(\omega_1^2 + \omega_2^2 + \omega_3^2 + \omega_4^2\right) \end{bmatrix}^t$ is the total thrust force expressed in F_I frame;
- $F_d = diag(k_{ftx}, k_{fty}, k_{ftz})V_I$ is the air drag force (k_{fxi} the drag coefficients);
- $F_g = \begin{pmatrix} 0 & 0 & mg \end{pmatrix}^t$ is the gravitational force;
- $M_a = diag(k_{fax}, k_{fay}, k_{faz})v^2{}_b$ is aerodynamic friction torque (k_{fai} the fiction coefficients);
- $M_f = \begin{pmatrix} U_2 & U_3 & U_4 \end{pmatrix}^t$ are the total rolling, pitching and yawing torques.
- M_{gb} and M_{gp} are the gyroscopic torques.

Using (3) and (6), the QUAV model is given by:

$$\begin{cases} \dot{p} = a_1 qr - a_3 \bar{\Omega}_r q + a_2 p^2 + b_1 U_2 \\ \dot{q} = a_4 pr + a_6 \bar{\Omega}_r p + a_5 q^2 + b_2 U_3 \\ \dot{r} = a_7 pq + a_8 r^2 + b_3 U_4 \\ \dot{\phi} = p + S_\phi \tan_\theta q + C_\phi \tan_\theta r \\ \dot{\theta} = C_\phi q - S_\phi r \\ \dot{\psi} = \frac{S_\phi}{C_\theta} q + \frac{C_\phi}{C_\theta} r \\ \ddot{x} = a_9 \dot{x} + \frac{1}{m} u_x U_1 \\ \ddot{y} = a_{10} \dot{y} + \frac{1}{m} u_y U_1 \\ \ddot{z} = a_{11} \dot{z} - g + \frac{\cos\phi \cos\theta}{m} U_1 \end{cases} \tag{7}$$

In which :

$$a_1 = \frac{(I_{yy} - I_{zz})}{I_{xx}}, a_2 = -\frac{k_{fax}}{I_{xx}}, a_3 = -\frac{J_r}{I_{xx}}, a_4 = \frac{(I_{zz} - I_{xx})}{I_{yy}},$$
$$a_5 = -\frac{k_{fay}}{I_{yy}}, a_6 = \frac{J_r}{I_{yy}}, a_7 = \frac{(I_{xx} - I_{yy})}{I_{zz}}, a_8 = -\frac{k_{faz}}{I_{zz}}, a_9 = -\frac{k_{ftx}}{m},$$
$$a_{10} = -\frac{k_{fty}}{m}, a_{11} = -\frac{k_{ftz}}{m}, b_1 = \frac{l}{I_{xx}}, b_2 = \frac{l}{I_{yy}}, b_3 = \frac{1}{I_{zz}}$$

$$\begin{cases} u_x = \cos(\phi)\sin(\theta)\cos(\psi) + \sin(\psi)\sin(\phi) \\ u_y = \cos(\phi)\sin(\theta)\sin(\psi) - \sin(\phi)\cos(\psi) \end{cases} \tag{8}$$

Assuming that the QUAV performs a smooth movement variation.i.e. ϕ/ θ are very small, so (5) is simplified as :

$$\Omega = \nu_b. \tag{9}$$

With ν_b is the QUAV's velocity expressed in the body frame.

From (7),(9) and by choosing the state vector $X = \begin{bmatrix} \phi & \dot{\phi} & \theta & \dot{\theta} & \psi & \dot{\psi} & x & \dot{x} & y & \dot{y} & z & \dot{z} \end{bmatrix}^t = \begin{bmatrix} x_1 & x_2 & x_3 & x_4 & x_5 & x_6 & x_7 & x_8 & x_9 & x_{10} & x_{11} & x_{12} \end{bmatrix}^t.$

The QUAV's state-space modeling is given by:

$$\begin{cases} \dot{x}_1 = x_2 \\ \dot{x}_2 = a_1 x_4 x_6 + a_2 {x_2}^2 + a_3 \bar{\Omega}_r x_4 + b_1 U_2 \\ \dot{x}_3 = x_4 \\ \dot{x}_4 = a_4 x_2 x_6 + a_5 {x_4}^2 + a_6 \bar{\Omega}_r x_2 + b_2 U_3 \\ \dot{x}_5 = x_6 \\ \dot{x}_6 = a_7 x_2 x_4 + a_8 {x_6}^2 + b_3 U_4 \\ \dot{x}_7 = x_8 \\ \dot{x}_8 = a_9 x_8 + \frac{1}{m} u_x U_1 \\ \dot{x}_9 = x_{10} \\ \dot{x}_{10} = a_{10} x_{10} + \frac{1}{m} u_y U_1 \\ \dot{x}_{11} = x_{12} \\ \dot{x}_{12} = a_{11} x_{12} - g + \frac{\cos(\phi)\cos(\theta)}{m} U_1 \end{cases} \tag{10}$$

3.2 Controller Scheme

The full control scheme is proposed as a multi-loop, Fig. 3. It is consisting of an inner loop that controls attitude and an outer loop to control the yaw and the translation.

The inner loop references are generated by the outer loop using the following equations:

$$\begin{cases} \phi_{ref} = \arcsin(u_x \sin(\psi_d) - u_y \cos(\psi_d)) \\ \theta_{ref} = \arcsin\left(\frac{u_x \cos(\psi_d) + u_y \sin(\psi_d)}{\cos(\phi_d)}\right) \end{cases} \tag{11}$$

Regarding the outer loop, the position references are obtained from the waypoints and the search trajectory during the crossing and searching phases. While for the tacking phase, the position and velocity errors are obtained by vision (section 5).

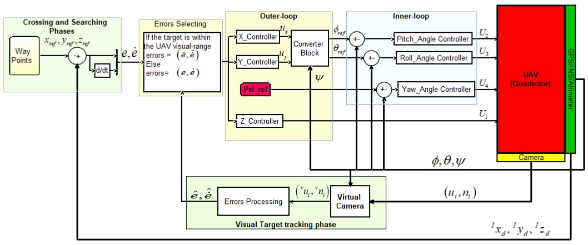

Figure 3: Proposed Controller Scheme.

Concerning the control mode, an SMC based on **E**xponential **R**eaching **L**aw (ERL) is adopted due to its robustness, low chattering effect and response time.

3.3 Sliding Mode Controller Design

The considered system (10) is a second order, so the sliding surface is given by:

$$S = \dot{e} + \lambda e \tag{12}$$

For attitude control, $e \in \{e_\phi, e_\theta\}$, $\dot{e} \in \{\dot{e}_\phi, \dot{e}_\theta\}$, $e_\phi = e_1 = \phi^d - \phi$, $e_\theta = e_3 = \theta^d - \theta$.

ϕ^d/θ^d are the desired QUAV's attitude (46) and ϕ/θ are the instantaneous QUAV's attitude obtained by IMU.

For the yaw control, $e_\psi = e_5 = \psi^d - \psi$, with ψ^d is the desired yaw, assumed to be constant and ψ the instantaneous yaw obtained by IMU.

Concerning the horizontal and vertical position control, the position errors (e_x, e_y, e_z) and their derivatives $(\dot{e}_x, \dot{e}_y, \dot{e}_z)$ are not accessible since the QUAV's position/velocity references are those of the uncooperative target.

Hence, the visual measurement detailed in (39), (44) and (48) are used to define the relative sliding surfaces:

$$S_n = \hat{\dot{e}}_n + \lambda_n \hat{e}_n, \; n \in \{x, y, z\} \tag{13}$$

In order to design the controllers, an approximate model called "control model" (14) is used, rather than defined in (10), in which the parameters uncertainty

$(\tilde{a}_i, \tilde{b}_i)$, the measurements noise $(\tilde{x}_i)$ and the external perturbations $(d_\bullet)$ are considered.

$$\begin{cases} \dot{\hat{x}}_1 = \tilde{x}_2 \\ \dot{\hat{x}}_2 = \tilde{a}_1\tilde{x}_4\tilde{x}_6 + \tilde{a}_2\tilde{x}_2^2 + \tilde{b}_1 U_2 + d_\phi \\ \dot{\hat{x}}_3 = \tilde{x}_4 \\ \dot{\hat{x}}_4 = \tilde{a}_4\tilde{x}_2\tilde{x}_6 + \tilde{a}_5\tilde{x}_4^2 + \tilde{b}_2 U_3 + d_\theta \\ \dot{\hat{x}}_5 = \tilde{x}_6 \\ \dot{\hat{x}}_6 = \tilde{a}_7\tilde{x}_2\tilde{x}_4 + \tilde{b}_3 U_4 + d_\psi \\ \dot{\hat{x}}_7 = \tilde{x}_8 \\ \dot{\hat{x}}_8 = \tilde{a}_9\tilde{x}_8 + \frac{1}{m}u_x U_1 + d_x \\ \dot{\hat{x}}_9 = \tilde{x}_{10} \\ \dot{\hat{x}}_{10} = \tilde{a}_{10}\tilde{x}_{10} + \frac{1}{m}u_y U_1 + d_y \\ \dot{\hat{x}}_{11} = \tilde{x}_{12} \\ \dot{\hat{x}}_{12} = \tilde{a}_{11}\tilde{x}_{12} - g + \frac{\cos(\phi)\cos(\theta)}{m}U_1 + d_z \end{cases} \tag{14}$$

With $\tilde{x}_i = x_i + noise$ are the noisy measurement . $d_\bullet$, $\bullet \in \{\phi, \theta, \psi, x, y, z\}$, with $|d_\bullet| \leq d_\bullet^{\max}$ are the external disturbances.

For simplification sake, the following expression of (14) is adopted:

$$\begin{cases} \dot{\tilde{x}}_i = \tilde{x}_{i+1} \\ \dot{\tilde{x}}_{i+1} = \tilde{f}_i(x) + \tilde{g}_i(x)U + d_\bullet \\ e_i = x_i^d - x_i \end{cases} \tag{15}$$

$i \in \{1,3,5,7,9,11\}$, $\mathrm{U} \in \{U_1, U_2, U_3, U_4, u_x, u_y\}$, $e_7 = e_x = x^d - x$, $e_9 = e_y = y^d - y$ and $e_{11} = e_z = z^d - z$
From (15), the first time derivative of the sliding surface is given by:

$$\dot{S} = \ddot{x}_i^d - \tilde{f}(\tilde{x}) - \tilde{g}(\tilde{x}).U - d_\bullet + \lambda.\dot{e} \tag{16}$$

Using (16), the general form of the control law is given by:

$$U = \frac{1}{\tilde{g}(x)}\left(\ddot{x}_i^d - \tilde{f}(x) + D_\bullet sign(S) + \lambda.\dot{e} - \dot{S}\right) \tag{17}$$

The term $D_\bullet sign(S)$ is introduced to compensate the external disturbance $d_\bullet$, such that:

$$d_\bullet^{\max} < D_\bullet \tag{18}$$

Defining the Lyapunov function :

$$V = \frac{1}{2}S^2 \tag{19}$$

The first time derivative of V must be negative defined to ensure the convergence,so:

$$\dot{V} = S\dot{S} < 0 \tag{20}$$

For ERL-based SMC, $\dot{S}$ is chosen as:

$$\dot{S} = -Ksign(S) - qS, q > 0, K > 0 \tag{21}$$

Replacing (21) in (19), the control law is given by:

$$U = \frac{1}{\tilde{g}(x)}\left(\ddot{x}_i^d - \tilde{f}(x) + D_\bullet sign(S) + \lambda.\dot{e} + Ksign(S) + qS\right) \tag{22}$$

This law forces the switching variable to reach the switching surface at a constant rate K, but if K is too small, the response time is important, on another hand, a large K will cause a severe chattering. By adding the proportional rate term $Ksign(S)$, the state is forced to approach switching manifold faster when the sliding surface S is larger. (Liu, 2017).

For uncooperative target tracking, the desired accelerations $\ddot{x}^d, \ddot{y}^d, \ddot{z}^d$ are those of the target, which are neither measurable nor estimable. Consequently, they can be considered as a pounded perturbation, such that:

$$|\ddot{x}_{(i)}^d| < \ddot{x}_{(i)\max}^d = \partial_i \tag{23}$$
$$\partial_i > 0, i \in \{5,7,9,11\}$$

Hence, the general expression of the outer loop controller is given by:

$$U = \frac{1}{\tilde{g}(x)}\left(-\tilde{f}(x) + \bar{D}_\bullet sign(S) + \lambda.\dot{e} - \dot{S}\right) \tag{24}$$

In this case, the term $\bar{D}_\bullet sign(S)$ compensates the external disturbance $d_\bullet$ and the unknown acceleration $\ddot{x}_{(i)}^d$. Therefore, the constant $\bar{D}_\bullet$ must satisfy the following condition:

$$d_\bullet^{\max} + \partial_i \leq \bar{D}_\bullet \tag{25}$$

Thus, considering $\dot{S}$ as given in (24), the different control laws are summarized as follows:

$$\begin{aligned} U_2 &= \frac{1}{\tilde{b}_1}\left[\dot{S}_\theta + \lambda_\theta.\dot{e}_\theta + \ddot{\theta}^d - \tilde{a}_4\tilde{x}_2\tilde{x}_6 - \tilde{a}_5\tilde{x}_4^2 + D_\theta sign(S_\theta)\right] \\ U_3 &= \frac{1}{\tilde{b}_2}\left[\dot{S}_\phi + \lambda_\phi.\dot{e}_\phi + \ddot{\theta}^d - \tilde{a}_1\tilde{x}_4\tilde{x}_6 - \tilde{a}_2\tilde{x}_2^2 + D_\phi sign(S_\phi)\right] \\ U_4 &= \frac{1}{\tilde{b}_3}\left[\dot{S}_\psi + \lambda_\psi.\dot{e}_\psi + \ddot{\psi}^d - \tilde{a}_7\tilde{x}_2\tilde{x}_4 - \tilde{a}_8\tilde{x}_6^2 + D_\psi sign(S_\psi)\right] \\ U_1 &= \frac{m}{\cos(\phi)\cos(\theta)}\left[\dot{S}_z + \lambda_z.\dot{\hat{e}}_z - \tilde{a}_{11}\tilde{x}_{12} + g + \bar{D}_z sign(S_z)\right] \\ u_y &= \frac{m}{U_1}\left[\dot{S}_y + \lambda_y.\dot{\hat{e}}_y - \tilde{a}_{10}\tilde{x}_{10} + \bar{D}_y sign(S_y)\right] \\ u_x &= \frac{m}{U_1}\left[\dot{S}_x + \lambda_x.\dot{\hat{e}}_x - \tilde{a}_9\tilde{x}_8 + \bar{D}_x sign(S_x)\right] \end{aligned} \tag{26}$$

Proof. Considering the expression of the control model (14), then substituting (16) in the time derivation of the Lyapunov function, we get:

$$\dot{V} = S\left(\ddot{x}_i^d - \tilde{f}(\tilde{x}) - \tilde{g}(\tilde{x}).U - d + \lambda.\dot{e}\right) \tag{27}$$

For the quadcopter's translation and yaw, replacing the controller U by its expression given by (24) with considering the $\dot{S}$ expression given in (21) , we get :

$$\dot{V} = \left(S\left(d + \ddot{x}_i^d\right) - K|S| - \bar{D}_\bullet|S| - qS^2\right) \tag{28}$$

When: $S < 0$, (28) become :

$$\dot{V} = -\left(|S|\left(\ddot{x}_i^d + d\right) + K|S| + \bar{D}_\bullet|S| + qS^2\right)$$

Under the condition (25), $q > 0$ and $K > 0$, it is clear that:

$$\dot{V} < 0 \tag{29}$$

When $S \geq 0$, from (28) we get:

$$S\left(d+\dot{x}_i^d\right) - \bar{D}_\bullet |S| \leq -\varepsilon, \varepsilon \geq 0$$

Hence:

$$\dot{V} < \left(-\varepsilon - \bar{D}_\bullet |S| - qS^2\right) < 0 \tag{30}$$

For the attitude control (ϕ and θ), the input control U of (27) is replaced by the expression given in (17) and following the same logic adopted for the translation's stability proof, with taking in account condition (18) instead of (25), the stability can be proved. □

4 VISION MODELS DERIVATION

The vision modeling involves, the camera modeling and the Camera Coverage Area (CCA) modeling.

4.1 Camera Modeling

The pinhole model is adopted in this work due to its simplicity and efficiency. It goes through three elementary and successive transformations (Fig .4).

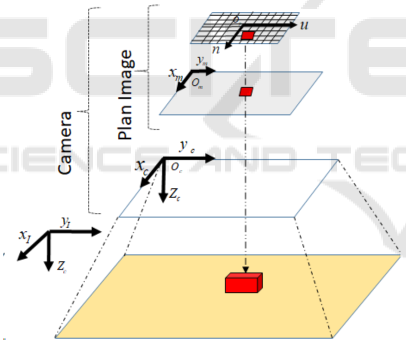

Figure 4: Projections for pinhole model.

Transformation 1: to express the coordinates of the points characterizing the target ${}^IP_i = \left({}^Ix_{p_i} \quad {}^Iy_{p_i} \quad {}^Iz_{p_i} \right)^t$ in the camera's frame.

$${}^cP_i = R_C \left({}^IP_i - {}^IO_c\right). \tag{31}$$

With ${}^IO_i \in \mathfrak{R}^3$ is the camera's position in F_I frame. $R_C = R^t$ is the rotation matrix between the F_I and F_C frames.

Transformation 2: is a perspective projection, transforming point ${}^cP_i$ into a two-dimensional point in the image plane $p_i = \left(x_{m_i} \quad y_{m_i} \right)^t$.

$$\begin{cases} x_{m_i} = f\frac{{}^cx_{p_i}}{{}^cz_{p_i}} \\ y_{m_i} = f\frac{{}^cy_{p_i}}{{}^cz_{p_i}} \end{cases} \tag{32}$$

With f is the camera's focal length.
Transformation 3: for the transformation from a metric positioning to a pixel positioning

$$\begin{cases} u_i = k_x x_{m_i} + u_0 \\ n_i = k_y y_{m_i} + n_0 \end{cases} \tag{33}$$

With (u_0, n_0) and k_x/k_y are the coordinates of the image's center and the number of pixels per unit of measurement respectively.

4.2 Camera Coverage Area Modeling

The CCA is a calculated measure that defines theoretically the maximum visible region from a camera. In the case of a rectangular modeling (Based on geometrical calculation) the CCA is limited by a rectangle centered in $O_{cca}\left({}^Ix_{cca}, {}^Iy_{cca}\right)$ and the maximal distances covered along x and y axis are given by (35) as depicted in Fig. 5 .

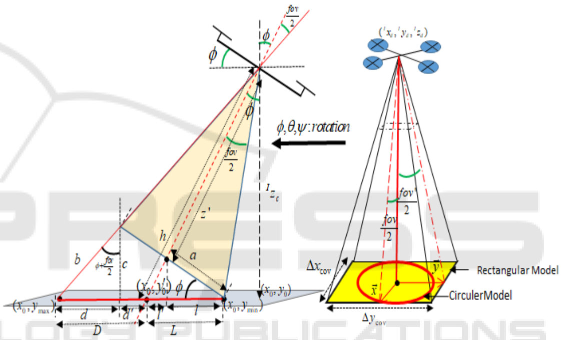

Figure 5: Camera Coverage Area Modeling.

$$\begin{cases} {}^Ix_{cca} = {}^Iz_c.T_{\frac{fov}{2}} \left(S_\theta T_{\theta + \frac{fov}{2}} + C_\theta - \frac{1}{C_\theta} \right) C_\psi - \\ {}^Iz_c.T_{\frac{fov'}{2}} \left(S_\phi T_{\phi + \frac{fov'}{2}} + C_\phi - \frac{1}{C_\phi} \right) S_\psi + {}^Iz_c.T_\theta + {}^Ix_c \\ {}^Iy_{cca} = {}^Iz_c.T_{\frac{fov}{2}} \left(S_\theta T_{\theta + \frac{fov}{2}} + C_\theta - \frac{1}{C_\theta} \right) S_\psi + \\ {}^Iz_c.T_{\frac{fov'}{2}} \left(S_\phi T_{\phi + \frac{fov'}{2}} + C_\phi - \frac{1}{C_\phi} \right) C_\psi + {}^Iz_c.T_\phi + {}^Iy_c \end{cases} \tag{34}$$

$$\begin{cases} \Delta x_{\text{cov}} = 2.{}^Iz_c.T_{\left(\frac{fov}{2}\right)} \left(S_{\theta_c} T_{\left(\theta_c + \frac{fov}{2}\right)} + C_{\theta_c} + \frac{1}{2C_{\theta_c}} \right) \\ \Delta y_{\text{cov}} = 2.{}^Iz_c.T_{\left(\frac{fov'}{2}\right)}. \left(S_{\phi_c} T_{\left(\phi_c + \frac{fov'}{2}\right)} + C_{\phi_c} + \frac{1}{2C_{\phi_c}} \right) \end{cases} \tag{35}$$

$T_\bullet = \tan(\bullet)$ and *fov*/*fov'* are the height and the width of the camera's field of view.
For a circular modeling, the model is a circle centered in C_0 with a rayon R_{cca}, such that:

$$\begin{cases} C_0 = \left({}^Ix_{cca}, {}^Iy_{cca}\right) \\ R_{cca} = \min(x_{\text{cov}}, y_{\text{cov}}) \end{cases} \tag{36}$$

For the detection model, the target is considered as automatically detected once it entered into the CCA.

5 TARGET POSITION AND VELOCITY ESTIMATION

A virtual camera with its associated virtual frame $F_v \equiv (o_c, \vec{e}^v_x, \vec{e}^v_y, \vec{e}^v_z)$ are introduced, so that the origin of the F_v coincides with the origin of the real camera's frame, and its $Z-axis$ $\vec{e}^v_z$ is aligned with the optical axis of the virtual camera, Fig 6.

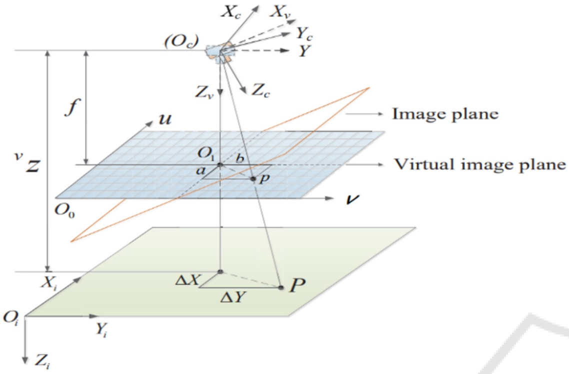

Figure 6: Virtual camera concept.

The virtual camera inherits only the yaw, and the horizontal translation of the real camera. By adopting this technique, we can claim that any changes in the image features are only due to the horizontal translation of the QUAV or to the target's movement. Therefore, the Horizontal Position's Errors (**HPE**) and the Velocity's Errors (**VE**) between the QUAV and the target are calculated using target image features expressed in the virtual camera. The same for the generation of the QUAV's relative altitude.

5.1 Horizontal Position's Errors Estimation

Considering a set of points $^cP_i \in \Re^3$ characterizing the target with their corresponding pixel coordinates $p_i = (u_i, n_i)$. Then, ^cP_i are projected in the virtual image plane as follows (Jabbari et al., 2012):

$$\begin{bmatrix} ^vu_i \\ ^vn_i \end{bmatrix} = \beta R^t_{\phi\theta} \begin{bmatrix} u_i \\ n_i \\ f \end{bmatrix} \tag{37}$$

$$\beta = f / \left(\begin{bmatrix} 0 & 0 & 1 \end{bmatrix} R^t_{\phi\theta} \begin{bmatrix} u_i \\ n_i \\ f \end{bmatrix} \right), R^t_{\phi\theta} = \left(R_\theta R_\phi\right)^t$$

Supposing that the target contains $N > 3$ non-collinear feature points, the image's feature centroid $^vp_g = (^vu_g, ^vn_g)$ in the virtual frame is given by:

$$\begin{cases} ^vu_g = \frac{1}{N} \sum\limits_{i=1}^{N} {}^vu_i \\ ^vn_g = \frac{1}{N} \sum\limits_{i=1}^{N} {}^vn_i \end{cases} \tag{38}$$

The idea of the control scheme is to drive the QUAV above the target, such that the desired virtual image's feature centroid is determined as the center of the virtual image plane $^vp_g = {}^vp_0$.
Hence, from equations (38) and (37), the horizontal position errors between the camera (QUAV) and the target are given by:

$$\begin{cases} \hat{e}_x = {}^v\hat{z}\frac{^vu_g - {}^vu_0}{f} \\ \hat{e}_y = {}^v\hat{z}\frac{^vn_g - {}^vn_0}{f} \end{cases} \tag{39}$$

Remark 1. *'$^v\hat{z}$' is the normal distance of the virtual camera from the target, and is assumed to be equal to the camera's vertical position $^I\hat{z}_c$.*

5.2 QUAV's Altitude Reference Generation

In a tracking scenario, the most intuitive and simple approach is to fly the QUAV at a fixed altitude, which must not be too high to ensure that the target is detectable on the image plane, and must also not be too low to ensure the visualization of all the target's details. However, this approach cannot guarantee that the QUAV flies at an optimal altitude. Thus, and to ensure this optimality, the technique proposed by (Zhang et al., 2020) to determine the desired QUAV's relative altitude is readopted in this work.
The proposed technique is summarized as follows:

- The introduction of a circle centered at O_c with radius r_{opt} as an optimal observation zone in the virtual image plane.
- Consider a circle centered at ^vp_g and passing through the farthest feature point of the target (projected in the virtual plane) to cover the whole target features. The circle radius r_{trg} is given by:

$$r_{trg} = \max\left\{ \sqrt{(^vu_i - {}^vu_g)^2 + \left((^vn_i - {}^vn_g)^2\right)} \right\}, i = 1,.,N \tag{40}$$

- Once the QUAV is above the target, its altitude will be controlled through r_{trg} which must be less than or equal to the r_{opt} to ensure that the target stays within the optimal viewing area.

$$r_d = \upsilon r_{opt}, \upsilon < 1. \tag{41}$$

υ is the radius-scaling factor.

Considering the perspective projection, we get:

$$r_d = f\frac{d_{trg}}{z_d} \tag{42}$$

d_{trg} is the unknown real distance between the target center and the farthest target feature point and z_d is the desired QUAV's altitude.

From (41) and (42) the desired vertical distance is given by:

$$z_d = r_{trg}\frac{{}^c\hat{z}_p}{\upsilon r_{opt}} \tag{43}$$

Therefore, the vertical error is given by:

$$\hat{e}_z = {}^c\hat{z}_p - z_d \tag{44}$$

In summary, the position errors are given by the following vector $\Delta\xi = \begin{pmatrix} \hat{e}_x & \hat{e}_y & \hat{e}_z \end{pmatrix}^t$.

5.3 Velocity Errors Estimation

To compute the velocity errors which is necessary for the controller. The expression of the target's feature points ${}^IP_i$ in the virtual frame is given by:

$${}^vP_i = R_\psi \left({}^IP_i - {}^IO_v\right) \tag{45}$$

Hence, the dynamic of the image features ${}^v\dot{p}_i = \begin{pmatrix} {}^v\dot{u}_i & {}^v\dot{n}_i \end{pmatrix}^t$ corresponding to the set point ${}^vP_i$ is obtained by the first time derivation of equation (45) considering that ${}^vz_{p_i} = {}^cz_{p_i} = {}^cz_p$.

$$\begin{bmatrix} {}^v\dot{u}_i \\ {}^v\dot{n}_i \end{bmatrix} = \frac{1}{{}^vz_{p_i}} \begin{bmatrix} f & 0 & -{}^vu_i \\ 0 & f & -{}^vn_i \end{bmatrix} \begin{bmatrix} {}^v\dot{x}_c - {}^v\dot{x}_{p_i} \\ {}^v\dot{y}_c - {}^v\dot{y}_{p_i} \\ {}^v\dot{z}_c - {}^v\dot{z}_{p_i} \end{bmatrix} + \begin{bmatrix} {}^vn_i \\ -{}^vu_i \end{bmatrix} \dot{\psi} \tag{46}$$

The image feature dynamic given by (46) is rewritten as follow:

$${}^v\dot{p}_i = {}^c\hat{z}_p^{-1} A_i {}^v\Delta v + B_i\dot{\psi} \tag{47}$$

Bearing in mind that the target feature points are coplanar and have the same target linear velocity, it is clear that in order to compute the three components of the velocity errors ${}^v\Delta v$, at least two points $(i \geq 2)$ are needed. Finlay, the velocity errors expressed in the F_I frame are given by:

$$\Delta v = {}^c\hat{z}_p R_\psi^t A^+ \left({}^v\dot{p} - B\dot{\psi}\right) \tag{48}$$

With $\Delta v = \begin{pmatrix} \hat{\dot{e}}_x & \hat{\dot{e}}_y & \hat{\dot{e}}_z \end{pmatrix}^t$, $A = \begin{pmatrix} A_i^t & .. & A_N^t \end{pmatrix}^t$, $Q = \begin{pmatrix} B_1^t & .. & B_N^t \end{pmatrix}^t$, A^+ is the Moore-Penrose pseudo-inverse of matrix A and ${}^v\dot{p} = \begin{pmatrix} {}^v\dot{p}_1^t & ... & {}^v\dot{p}_N^t \end{pmatrix}^t$ is obtained by measuring the optic flow.

5.4 QUAV's Relative Vertical Position Measurement

Given the geometric models of the possible targets, the horizontal distance between the camera and the target is calculated using the image coordinates of at least two target feature points and the corresponding real distance between, obtained from the model matching, them combined with equations (32) and (33). It is given by:

$${}^c\hat{z}_p = f.D_{i/j}\left(\left(\frac{u_i - u_j}{k_x}\right)^2 + \left(\frac{n_i - n_j}{k_y}\right)^2\right)^{-1/2} \tag{49}$$

$D_{i/j}$ is the horizontal distance between two points P_i/P_j on the target, obtained by model matching. u_i/n_i and u_j/n_i are their corresponding pixel coordinates.

6 SIMULATION AND DISCUSSION

For the numerical simulations, the QUAV's parameters uncertainty is chosen such that $\tilde{a}_i = a_i \pm 15\% a_i$ and $\tilde{b}_1 = b_i \pm 15\% b_i$.

The solver Runge-Kutta is used with a fixed-step (0.001 simpling rate).

Concerning the target, it is considered as a rectangular object, in which the visual features include its four vertexes with the following coordinates (in meters) $(0.2, 0.2, 0)$, $(-0.2, 0.2, 0)$, $(-0.2, -0.2, 0)$ and $(0.2, -0.2, 0)$.

The camera position and attitude with respect to F_B frame are $(0, 0, 0.05)\,m$ and $(0, 0, 0)\,rad$ respectively.

The QUAV's initial position and attitude are $(0, 0, 0.1)\,m$, $(0, 0, 0)\,rad$, respectively. The coordinates of the starting point of the search phase is $(8, 7, 0)\,m$. The crossing and searching altitudes are $5m$ and $20m$ respectively. The searching trajectory is a circle centered in the searching start point with a rayon equal to R_{CCa}.

The QUAV's parameters uncertainty is chosen such that $\tilde{a}_i = a_i \pm 15\% a_i$ and $\tilde{b}_1 = b_i \pm 15\% b_i$. To evaluate the visual measurement for the three flight phases, we consider that the target is always visible and the tracking process is started at the instant $t = 55s$.

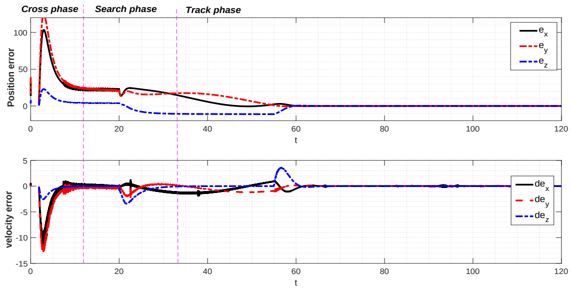

Figure 7: Visual measurement: position and velocity errors (with sign function).

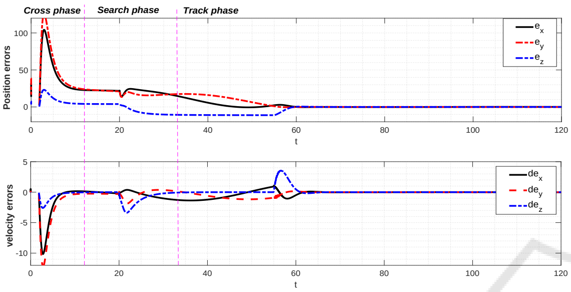

Figure 8: Visual measurement: position and velocity errors.

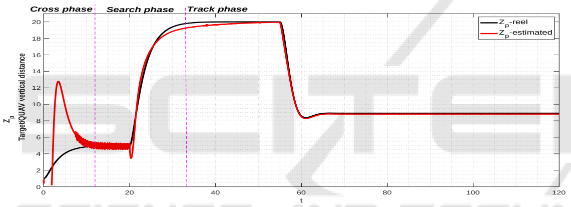

Figure 9: Simulation results of the proposed Vision-based vertical distance measurement (with sign function).

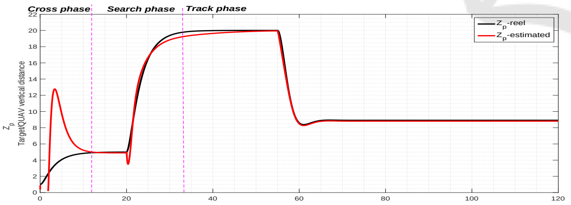

Figure 10: Simulation results of the proposed Vision-based vertical distance measurement.

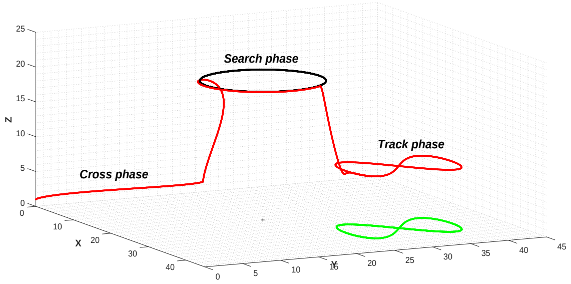

Figure 11: Simulation results of the proposed flight scenario in $3D$.

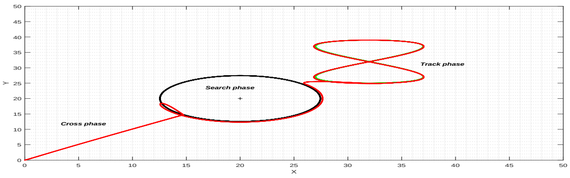

Figure 12: Simulation results of the proposed flight scenario in $2D$.

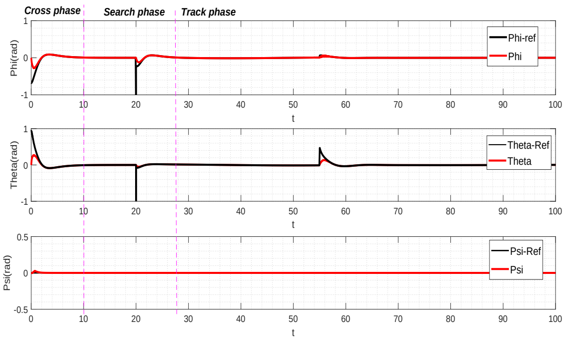

Figure 13: Simulation results of the QUAV's attitude and yaw control.

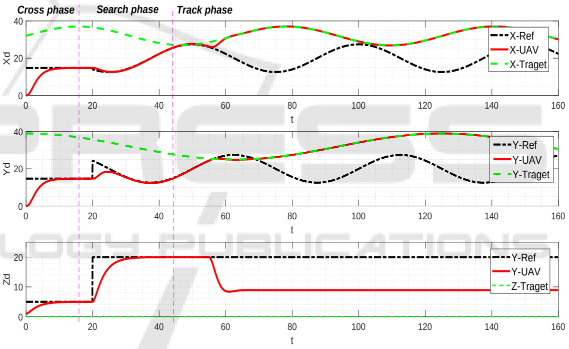

Figure 14: Simulation results of the QUAV's translation movement control.

The simulation was performed by considering the saturation function instead of the sign function in the proposed controller, since the use of the saturation function lead to a considerable improvement in the visual measurement results (Fig. 8 and Fig. 10), unlike the sign function, which gave a noisy result, (Fig. 7 and Fig. 9).

The flight scenario is depicted in Fig. 11 and Fig. 12 (3D and 2D), in which the three flight phases are illustrated: the cruise phase, the search phase and the tracking phase. In Fig. 11 the automatic altitude tuning is clearly depicted, where the QUAV has reduced its altitude automatically for vision optimization.

For the control law results, Fig. 14 shows the control results of the QUAV's horizontal and vertical trajectory. Figure. 13 exposes the QUAV's attitude and yaw control, where the chattering effect is too low.

7 CONCLUSION

The proposed strategy for uncooperative mobile ground target tracking using a quadcopter has been assessed using simulated scenarios. The evolved IBVS approach allows improving searching, detection and tracking efficiency while the proposed ERL based Sliding mode controller guarantees the stability and robustness of the QUAV. A simple searching law was used for detecting a moving target and its relative position and velocity estimation considering uncertainties. Simulations were performed to successfully demonstrate the performance and feasibility of the proposed method.

REFERENCES

Borshchova, I. and O'Young, S. (2016). Visual servoing for autonomous landing of a multi-rotor uas on a moving platform. *Journal of Unmanned Vehicle Systems*, 5(1):13–26.

Cantelli, L., Mangiameli, M., Melita, C. D., and Muscato, G. (2013). Uav/ugv cooperation for surveying operations in humanitarian demining. In *2013 IEEE international symposium on safety, security, and rescue robotics (SSRR)*, pages 1–6. IEEE.

Cao, Z., Chen, X., Yu, Y., Yu, J., Liu, X., Zhou, C., and Tan, M. (2017). Image dynamics-based visual servoing for quadrotors tracking a target with a nonlinear trajectory observer. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 50(1):376–384.

Chaumette, F. and Hutchinson, S. (2006). Visual servo control. i. basic approaches. *IEEE Robotics & Automation Magazine*, 13(4):82–90.

Ding, X. C., Rahmani, A. R., and Egerstedt, M. (2010). Multi-uav convoy protection: An optimal approach to path planning and coordination. *IEEE transactions on Robotics*, 26(2):256–268.

Duggal, V., Sukhwani, M., Bipin, K., Reddy, G. S., and Krishna, K. M. (2016). Plantation monitoring and yield estimation using autonomous quadcopter for precision agriculture. In *2016 IEEE international conference on robotics and automation (ICRA)*, pages 5121–5127. IEEE.

Fink, G., Xie, H., Lynch, A. F., and Jagersand, M. (2015). Experimental validation of dynamic visual servoing for a quadrotor using a virtual camera. In *2015 International conference on unmanned aircraft systems (ICUAS)*, pages 1231–1240. IEEE.

Jabbari, H., Oriolo, G., and Bolandi, H. (2012). Dynamic ibvs control of an underactuated uav. In *2012 IEEE International Conference on Robotics and Biomimetics (ROBIO)*, pages 1158–1163. IEEE.

Janabi-Sharifi, F. and Marey, M. (2010). A kalman-filter-based method for pose estimation in visual servoing. *IEEE transactions on Robotics*, 26(5):939–947.

Liu, J. (2017). *Sliding mode control using MATLAB*. Academic Press.

Menouar, H., Guvenc, I., Akkaya, K., Uluagac, A. S., Kadri, A., and Tuncer, A. (2017). Uav-enabled intelligent transportation systems for the smart city: Applications and challenges. *IEEE Communications Magazine*, 55(3):22–28.

Pestana, J., Sanchez-Lopez, J. L., Saripalli, S., and Campoy, P. (2014). Computer vision based general object following for gps-denied multirotor unmanned vehicles. In *2014 American Control Conference*, pages 1886–1891. IEEE.

Prabhakaran, A. and Sharma, R. (2021). Autonomous intelligent uav system for criminal pursuit–a proof of concept. *The Indian Police Journal*, page 1.

Puri, A. (2005). A survey of unmanned aerial vehicles (uav) for traffic surveillance. *Department of computer science and engineering, University of South Florida*, pages 1–29.

Radiansyah, S., Kusrini, M., and Prasetyo, L. (2017). Quadcopter applications for wildlife monitoring; iop conference series: Earth and environmental science.

Samad, T., Bay, J. S., and Godbole, D. (2007). Network-centric systems for military operations in urban terrain: The role of uavs. *Proceedings of the IEEE*, 95(1):92–107.

Wang, Z., Gao, Q., Xu, J., and Li, D. (2022). A review of uav power line inspection. *Advances in Guidance, Navigation and Control*, pages 3147–3159.

Zhang, K., Shi, Y., and Sheng, H. (2021). Robust nonlinear model predictive control based visual servoing of quadrotor uavs. *IEEE/ASME Transactions on Mechatronics*, 26(2):700–708.

Zhang, S., Zhao, X., and Zhou, B. (2020). Robust vision-based control of a rotorcraft uav for uncooperative target tracking. *Sensors*, 20(12):3474.