Quadrotor UAV Dynamic Visual Servoing Based on Differential Flatness Theory

Ahmed Alshahir, Mohammed Albekairi, Kamel Berriri, Hassen Mekki, Khaled Kaaniche, Shahr Alshahr, Bassam Alshammari, Anis Sahbani

applied sciences

MDPI

Article

Quadrotor UAV Dynamic Visual Servoing Based on Differential Flatness Theory

Ahmed Alshahir [1,*], Mohammed Albekairi [1], Kamel Berriri [2], Hassen Mekki [3], Khaled Kaaniche [1], Shahr Alshahr [1], Bassam A. Alshammari [1] and Anis Sahbani [4]

[1] Department of Electrical Engineering, College of Engineering, Jouf University, Sakaka 72388, Saudi Arabia
[2] LAMMDA Laboratory, University of Sousse, Sousse 4054, Tunisia
[3] National School of Engineering of Sousse, NOCCS Laboratory, University of Sousse, Sousse 4054, Tunisia
[4] Institute for Intelligent Systems and Robotics (ISIR), CNRS, Sorbonne University, 75006 Paris, France
* Correspondence: aaalshahir@ju.edu.sa

Abstract: In this paper, we propose 2D dynamic visual servoing (Dynamic IBVS), where a quadrotor UAV tries to track a moving target using a single facing-down perspective camera. As an application, we propose the tracking of a car-type vehicle. In this case, data related to the altitude and the lateral angles have no importance for the visual system. Indeed, to perform the tracking, we only need to know the longitudinal displacements (along the x and y axes) and the orientation along the z-axis. However, those data are necessary for the quadrotor's guidance problem. Thanks to the concept of differential flatness, we demonstrate that if we manage to extract the displacements according to the three axes and the orientation according to the yaw angle (the vertical axis) of the quadrotor, we can control all the other variables of the system. For this, we consider a camera equipped with a vertical stabilizer that keeps it in a vertical position during its movement (a gimbaled camera). Other specialized sensors measure information regarding altitude and lateral angles. In the case of classic 2D visual servoing, the elaboration of the kinematic torsor of the quadrotor in no way guarantees the physical realization of instructions, given that the quadrotor is an under-actuated system. Indeed, the setpoint has a dimension equal to six, while the quadrotor is controlled only by four inputs. In addition, the dynamics of a quadrotor are generally very fast, which requires a high-frequency control law. Furthermore, the complexity of the image processing stage can cause delays in motion control, which can lead to target loss. A new dynamic 2D visual servoing method (Dynamic IBVS) is proposed. This method makes it possible to generate in real time the necessary movements for the quadrotor in order to carry out the tracking of the target (vehicle) using a single point of this target as visual information. This point can represent the center of gravity of the target or any other part of it. A control by flatness has been proposed, which guarantees the controllability of the system and ensures the asymptotic convergence of the generated trajectory in the image plane. Numerical simulations are presented to show the effectiveness of the proposed control strategy.

Keywords: target and feature tracking; autonomous unmanned aerial vehicles; quadrotor; dynamic image-based visual servoing; flatness control

Citation: Alshahir, A.; Albekairi, M.; Berriri, K.; Mekki, H.; Kaaniche, K.; Alshahr, S.; Alshammari, B.A.; Sahbani, A. Quadrotor UAV Dynamic Visual Servoing Based on Differential Flatness Theory. *Appl. Sci.* **2023**, *13*, 7005. https://doi.org/10.3390/app13127005

Academic Editors: Leszek Ambroziak and Cezary Kownacki

Received: 11 May 2023
Revised: 31 May 2023
Accepted: 5 June 2023
Published: 10 June 2023

1. Introduction

The navigation of unmanned aerial vehicles (UAVs) using a vision system has found much interest during the last few decades in several fields of its application, such as the military field and civil society [1,2], traffic surveillance [3,4], mapping and exploration [5,6], and agriculture [7,8]. Visual servoing methods use visual information to control a vehicle's pose relative to specific visual targets. They are divided into two main families [9]: 3D visual servoing, or position-based visual servoing (PBVS), and 2D visual servoing, or image-based visual servoing (IBVS). IBVS directly uses image feature errors in the image plane to derive control inputs. It regulates the vehicle without reconstructing the relative

pose with respect to visual targets. It therefore does not need information on the geometry of the target a priori, as in the case of PBVS. Moreover, it is easy to calculate and more robust than PBVS.

The IBVS technique is a control method that guarantees the convergence of the visual features of a target toward the desired values in the image, as stated in [9]. IBVS methods may face challenges, such as significant tracking inaccuracies or total tracking failure, in situations where the motion of the target changes over time or is not accurately anticipated [10]. Predictive visual control (PVC) tries to solve this problem by incorporating model predictive control constraints [11,12]. These constraints include the field of view (FOV), actuator output limitations, and the workspace. In [13], a nonlinear predictive controller was effectively employed to produce the desired velocity for an underwater vehicle while adhering to visibility limitations. The same approach was investigated in [14] to develop a tracking controller for UAVs. The application of the model predictive control (MPC) scheme has been observed in the context of a mobile robot [15] and quadrotor [16]. In the aforementioned scenario, the model predictive control (MPC) was employed to ensure that the visual attribute of the target remains in the intended location within the image. In the context of navigation, the IBVS system has been observed to encounter occlusion issues leading to missing feature points. To address this, artificial patterns have been utilized to predict the missing feature points and maintain the proper functioning of the system. This approach has been documented in [17]. Nevertheless, the model predictive control (MPC) methodologies are restricted to immobile targets. Furthermore, a high-performance processing stage is necessary. Predictive control is a computational process that involves solving an optimization problem at every moment in real time. This is due to the fact that predictive control requires intensive calculations. This situation may result in a substantial computational load, particularly for complex or fast systems, as occurs in our scenario.

For effective tracking control, it is crucial to have knowledge of the movement of a dynamic target. This information is frequently inaccessible and challenging to anticipate. Closed-loop control employs various image characteristics to maintain the target within the field of view (FOV), as stated in [18]. Various techniques have been developed for feature extraction and matching in image processing. These include RGB-based methods [19], scale-invariant feature transformation (SIFT) [20], and accelerated robust features (SURF) [21]. Notwithstanding, these techniques exhibit constraints with respect to object detection and the assessment of the camera's motion relative to the target. Quadrotors have been subjected to vision-based optimization techniques [22] for the purpose of tracking a moving target while avoiding obstacles. However, this is only possible if the target's position is predetermined. In [23], alternative model-based optimization techniques were employed to ensure reliable detection of an unmanned aerial vehicle (UAV), but the focus of the study was primarily on utilizing image features for indoor localization instead of target tracking. Several methods have been employed to track humans, including the use of bounding boxes and minutiae [24,25]. Nevertheless, the targets tracked using these methods moved at slow speeds, which could result in their movements being ignored. Furthermore, it has been reported that target orientation is frequently unavailable [26]. In [27], the utilization of model-based predictive control was demonstrated for the purpose of tracking a periodically moving target. This approach resulted in a reduction in the complexity associated with controller design. The aforementioned methods failed to consider the interaction between the unmanned aerial vehicle (UAV) and the intended target. Additionally, the angle between the camera and the target was neither modeled nor quantified.

Quadrotor dynamics are typically fast and unpredictable. To control this type of system, it is necessary to develop a high-frequency controller [28]. On the other hand, visual servoing goes through an image processing step that aims to extract the characteristics of the object. This can have a detrimental effect on the frequency of control law development. Furthermore, the complexity of image processing can cause delays in motion control, which can lead to the loss of a target. In order to solve these problems, a new 2D dynamic visual servoing (Dynamic IBVS) method is proposed. Its objective is to generate the necessary

movements of the quadrotor to keep the target centered in the image plane. The proposed method transforms the problem into an asymptotic tracking process of a desired trajectory in the image plane, using the inverse dynamic of the estimated model of the vehicle to be followed. Since the proposed method allows the altitude of the quadrotor to be controlled independently of other variables, it is possible to set the altitude to a high level in order to reduce the risk of losing the target out of sight of the on-board camera, even during discontinuous and significant movements of the target. Moreover, this method only uses a single point on the target as a visual primitive. To increase the robustness and flexibility of the detection, this point can represent either the center of gravity of the target to be tracked or a specific part of the target.

The flatness property of a system is a relatively recent concept in automatic control that was proposed and developed in 1992 by M. Fliess et al. [29]. This property, which makes it possible to parameterize in a very simple way the dynamic behavior of a system, is based on the highlighting of a set of fundamental variables of the system: its flat outputs. This point of view, as we demonstrate, has multiple and interesting consequences relative to the control of systems. First of all, this makes it possible to return to the center of the control of a process the notion of trajectory that the system must execute; that is to say, the movement requested from a system must above all be achievable by this system. This avoids many of the problems faced by automation engineers. One of the first steps in flatness control is to generate an adequate desired trajectory that implicitly takes into account the system model.

In this work, we consider as an application the tracking of a car-type vehicle (Dynamic IBVS) by a quadrotor UAV equipped with a single facing-down perspective camera. In our case, the information concerning the altitude and the lateral angles (the roll angle and the pitch angle) are of no importance to the visual system. Indeed, to perform the tracking, we only need to know the longitudinal displacements (along the x- and y-axes) and the orientation along the z-axis. Those details are necessary for the problem of guiding the quadrotor. In [30–33], the authors proposed to use a rotating image plane, called a "virtual image plane", thus making it possible to obtain a dynamic of decoupled image characteristics. This method is applied to a fixed target and requires the detection of at least three points on the target. Thanks to the concept of differential flatness, we demonstrate that if we manage to extract the displacements according to the three axes and the orientation according to the yaw angle (the vertical axis) of the quadrotor, we can control all the other variables of the system. For this, we consider the following conditions. The camera is equipped with a vertical stabilizer, which keeps the camera in a vertical position during its movement; in other words, we neglect the lateral angles. It is also assumed that the quadrotor flies over at a given altitude. This altitude is not necessarily constant, but it must be known a priori. It should be noted here that these hypotheses only concern the visual system, which makes it possible to generate the movements necessary for the quadrotor in order to ensure the tracking of the vehicle. We use additional sensors to measure its magnitudes in order to achieve the trajectory that the visual system has thus generated.

In the case of traditional 2D visual servoing, the development of the quadrotor's kinematic torsor does not guarantee the physical realization of control instructions (a controllability issue compounded by an under-actuated system). In fact, the kinematic torsor has six dimensions, whereas the quadrotor has only four inputs. With only four control inputs, it is nearly impossible to implement the six instructions generated by the visual servoing algorithm. To solve this problem, ref. [34] proposed a linear model predictive control (MPC), but this method uses linear approximations and is not generally suitable for systems with very fast dynamics. Dongliang Zheng et al. [31] offered a command by backstepping; it was necessary to make many modifications to the model to render it in a particular form.

The proposed control by flatness takes into account all the variables of the system, guarantees its controllability, and ensures the asymptotic convergence of the resulting trajectory. The contributions of this study can be summarized as follows:

i. Using the concept of differential flatness, we have developed a new method of dynamic visual servoing for quadrotors. This method generates the necessary movements (translation and orientation) in order to keep the target centered in the image plane.
ii. Since quadrotors are fast systems working in outdoor environments, we have simplified the image processing and ensured the robustness of the visual primitive by using only one point of the target.
iii. Since quadrotors are under-actuated and strongly coupled systems, the realization of the kinematic tensor generated by the visual servoing algorithm becomes a problem. To solve this, we have proposed a control by flatness that ensures controllability and asymptotic tracking of the generated trajectory.
iv. In order to ensure robustness against climatic conditions, such as wind, we have added a PD-type correction term to the open-loop flatness control.

This paper is organized as follows: Section 2 presents the dynamic model of the quadrotor. The tracking strategy of a vehicle is detailed in Section 3. This strategy includes three loops: the first controls the altitude of the quadrotor; the second is dedicated to the generation of the trajectory; and finally, the third loop ensures tracking by flatness. Section 4 displays the simulation results validating the proposed approach.

2. The Dynamic Model of the Quadrotor

The commonly used quadrotor dynamic model [35–37] is given by Equation (1). This model has been proven by numerous experimental tests.

$$\begin{cases} \ddot{x} = u_1(cos\psi sin\theta cos\phi + sin\psi sin\phi) - \frac{K_1\dot{x}}{m} \\ \ddot{y} = u_1(sin\psi sin\theta cos\phi - cos\psi sin\phi) - \frac{K_2\dot{y}}{m} \\ \ddot{z} = u_1(cos\theta cos\phi) - g - \frac{K_3\dot{z}}{m} \\ \ddot{\theta} = u_2 - \frac{lK_4\dot{\theta}}{I_1} \\ \ddot{\phi} = u_3 - \frac{lK_5\dot{\phi}}{I_2} \\ \ddot{\psi} = u_4 - \frac{K_6\dot{\psi}}{I_3} \end{cases}, \quad (1)$$

where (x, y, z) are the three positions; (θ, ϕ, ψ) are the three Euler angles, representing pitch, roll, and yaw, respectively; g is the acceleration of gravity; l is the distance from the center of gravity to each rotor; m is the total mass of the quadrotor; (I_1, I_2, I_3) are the moments of inertia along x, y, and z; $(K_1, K_2, K_3, K_4, K_5, K_6)$ are the drag coefficients (in the rest of our work, we assume that the drag is zero since the drag is negligible at low speed); (u_1, u_2, u_3, u_4), are the control inputs defined Equation (2) [36]:

$$\begin{cases} u_1 = \frac{(T_1+T_2+T_3+T_4)}{m} \\ u_2 = \frac{l(-T_1-T_2+T_3+T_4)}{I_1} \\ u_3 = \frac{l(-T_1+T_2+T_3-T_4)}{I_2} \\ u_4 = \frac{C(T_1-T_2+T_3-T_4)}{I_3} \end{cases}, \quad (2)$$

where (T_1, T_2, T_3, T_4) are the thrusts generated by the four rotors and can be considered as the actual system control inputs; C is the force-moment scaling factor; u_1 represents the total thrust on the quadrotor UAV body according Z; u_2 and u_3 are the pitch and roll inputs; and u_4 is the yaw input.

3. Tracking Strategy

The control strategy of the quadrotor for ensuring the tracking of a vehicle is given in Figure 1. The quadrotor takes a desired altitude, z_d, and as soon as it detects the vehicle to be pursued, it will join it and ensure its tracking. The quadrotor used in this work is equipped with a camera with a stabilizer that keeps this camera upright during its

movement. Once we have generated the movements necessary for the quadrotor to ensure the tracking of the vehicle, a flatness control technique is proposed to carry out this target.

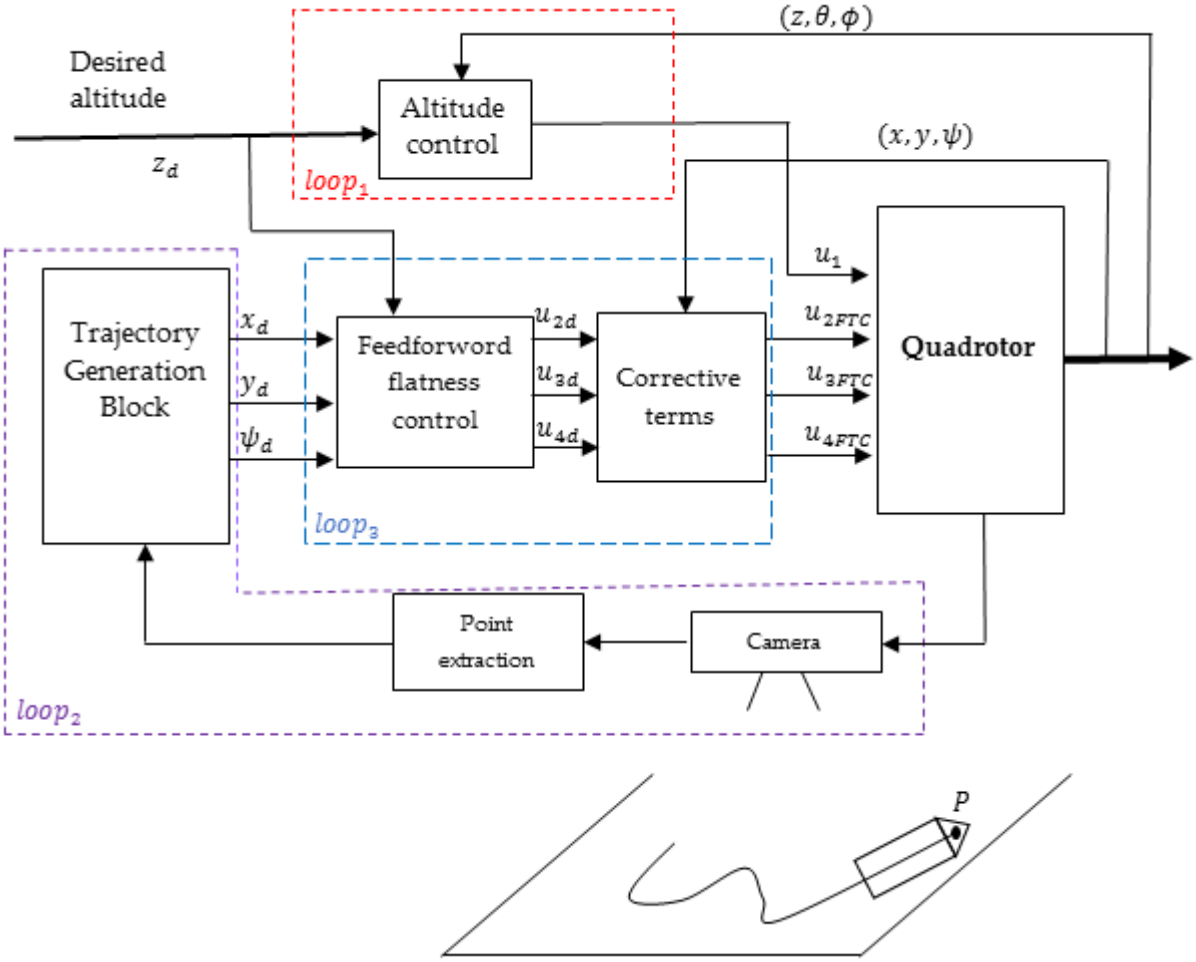


Figure 1. Vehicle tracking strategy.

We demonstrate in this work that a single point of the object to be tracked is enough for our proposed algorithm to achieve visual servoing. The trajectory generation block uses the coordinates in the image plane of this point to generate the necessary movements for the quadrotor in order to track the vehicle. In this control strategy, we develop three control loops, namely the loop that controls the altitude of the quadrotor; the loop that provides the 2D dynamic visual servoing, generating in real time the correct movements for the quadrotor in order to perform the tracking of the vehicle; and the loop that ensures the asymptotic convergence of the desired trajectory with a given degree of robustness using flatness control.

3.1. Loop 1: Altitude Control

As mentioned in Section 2, the control input, u_1, is responsible for movement along the Z-axis. By applying the input/output linearization method on the third line of Equation (1), the linearizing control will be given by:

$$\begin{cases} u_1 = \frac{(Nu_z + g)}{cos\theta cos\varphi} \\ with\ cos\theta cos\varphi \neq 0 \end{cases}, \tag{3}$$

where Nu_z is the new input of the linearized system given by Equation (4):

$$\ddot{z} = Nu_z. \tag{4}$$

To make the altitude, $z(t)$, track the desired altitude, $z_d(t)$, we just take the new input as follows:

$$Nu_z = \ddot{z}_d + k_{11}(\dot{z}_d - \dot{z}) + k_{12}(z_d - z). \quad (5)$$

The coefficients k_{11} and k_{12} are chosen so that the polynomial $p^2 + k_{11}p + k_{12}$ is a Hurwitz polynomial.

3.2. Loop 2: Trajectory Generation

By using a vertical camera stabilizer, the image plane will always be parallel to the (X_w, Y_w) plane of the Cartesian world coordinate system, as shown in Figure 2.

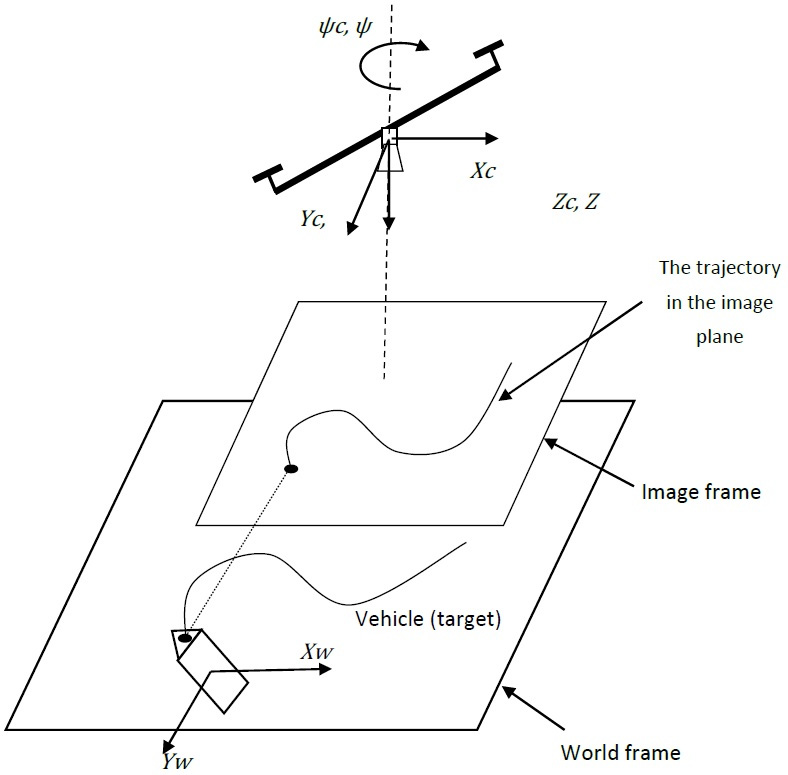


Figure 2. Quadrotor coordinate system vs. Cartesian world coordinate system.

We denote $(x, y, z, \theta, \phi, \psi)$ as quadrotor coordinates and $(x_c, y_c, z_c, \theta_c, \phi_c, \psi_c)$ as camera coordinates. So, we have $x_c = x$; $y_c = y$; $z_c = z$; $\psi_c = \psi$; $\theta_c = 0$; and $\varnothing_c = 0$. We assume that the quadrotor flies at a given constant altitude, z_d, and we impose that the orientation of the quadrotor is the same as that of the vehicle to be tracked (the orientation is managed by the Yaw angle $\psi_c = \psi$). The translation movements along the X_w- and the Y_w-axes as well as the rotation along the Z_w-axis of the quadrotor are independent. In other words, to go from an initial situation to a final situation, there is an infinity of possible trajectories. We are therefore faced with a problem that is undersized: we have three variables to determine, (X_d, Y_d, ψ_d) using only two equations (coordinates of the point, P, in the image plane). To remedy this problem, we choose a trajectory that connects the two situations in a way similar to that carried out by a differential mobile robot. This choice has legitimacy since we aim to track a car-type vehicle. We can therefore assimilate the behavior of the camera on board the quadrotor to that of a differential mobile robot that moves in the plane parallel to the plane (X_w, Y_w) located at a distance, z_d, from this plane and rotating according to the axis Z_w by an angle, ψ, according to the following dynamics:

$$\begin{cases} \dot{x}_r = v_r cos(\psi) \\ \dot{y}_r = v_r sin(\psi) \\ \dot{\psi} = \omega_r \end{cases}, \quad (6)$$

where $\dot{x}_r$ and $\dot{y}_r$ are the translation velocities along the X_w- and Y_w-axes of the robot. v_r and ω_r are, respectively, the linear speed and the angular speed of the robot.

3.2.1. Characteristics of the Descriptor

The tracking problem using a camera as a visual sensor is a 2D dynamic visual servoing problem. In this case, tracking is guaranteed if we manage to keep the vehicle (the target) centered in the image plane, as shown in Figure 3.

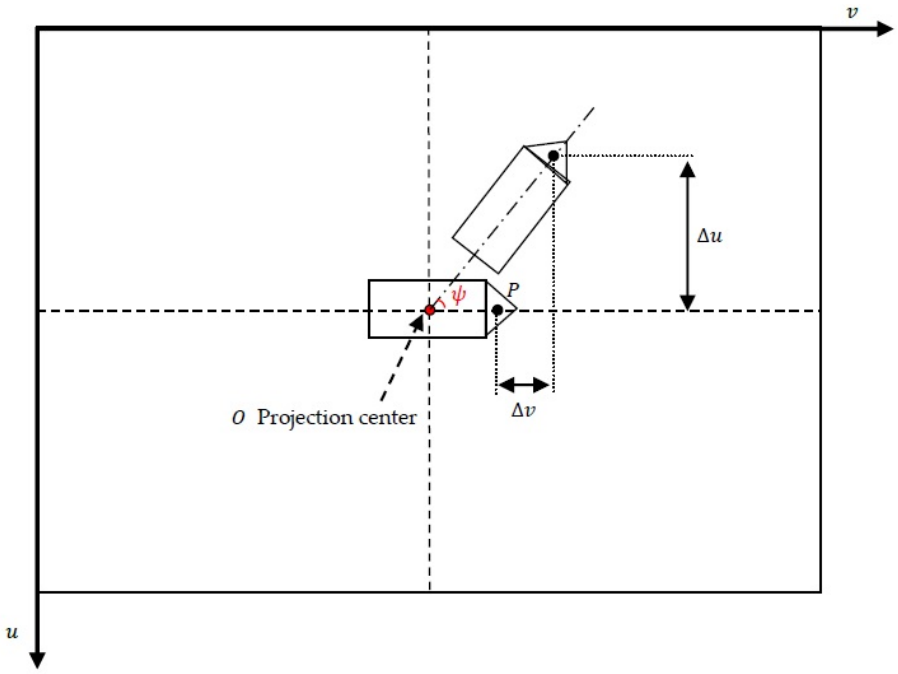


Figure 3. Image coordinate system.

According to the behavior of the dynamics proposed in Equation (6), the knowledge of the displacements $(\Delta x_r, \Delta y_r)$ according to the axes X_w and Y_w makes it possible to deduce the orientation ψ along the Z_w-axis. In effect,

$$\psi = atan\left(\frac{\Delta y_r}{\Delta x_r}\right). \tag{7}$$

Since we know $z_c = z = z_d$, we can deduce the displacements $(\Delta x_r, \Delta y_r)$ from the coordinates of a single point, P, of the vehicle (the target). This point can be chosen in an arbitrary way, belonging to the vehicle but other than that which coincides with the center of projection of the camera because it is invariant to the rotation according to the axis Z_w.

It is interesting here to take the point, P, on the horizontal axis, H, of the image (Figure 3) passing through the center of projection, o, because all the points located on this axis are invariant to displacement along Z_w. To simplify the task of extraction, this point can represent the center of gravity of the vehicle or even the center of gravity of the vehicle hood, as shown in Figure 3. Let $\boldsymbol{X} = (X_c, Y_c, Z_c = z_d = constant)$, the coordinates of the point, P, in the 3D Cartesian coordinate system; the projection of this point on the image plane is done in p of coordinates $(x_m,\ y_m)$ expressed in millimeters. The visual information considered in this work is $S = (x_m,\ y_m)$. The expressions of these coordinates are given by the following relations:

$$\begin{cases} x_m = \frac{X_c}{Z_c} = \frac{(u-c_u)}{f.\alpha_u} \\ y_m = \frac{Y_c}{Z_c} = \frac{(v-c_v)}{f.\alpha_v} \end{cases}, \tag{8}$$

where (u, v) represents the coordinates of the point p of the image expressed in pixels. $a = (c_u, c_v, f, \alpha_u, \alpha_v)$ is the set of intrinsic parameters of the camera with (c_u, c_v), the

coordinates of the principal point of the image; f is the focal length; and (α_u, α_v) are the vertical and horizontal scale factors expressed in pixel/mm. By performing the derivation with respect to the time of the projection equations in (8) we obtain:

$$\dot{\mathbf{S}} = l_s.V, \tag{9}$$

where V is the kinematic torsor of the camera formed by the translation velocities, v_c, and the rotation velocities, ω_c. L_s denotes the interaction matrix, also known as the Jacobian of the image, and is given by:

$$L_s = \begin{bmatrix} -\frac{1}{z_d} & 0 & \frac{x_m}{z_d} & x_m y_m & -(1+x_m^2) & y_m \\ 0 & -\frac{1}{z_d} & \frac{y_m}{z_d} & 1+y_m^2 & -x_m y_m & -x_m \end{bmatrix}. \tag{10}$$

Since the movement of the robot is assumed to be in a plane and, using Equation (7), we can conclude that if we know the translational speed along the X_c-axis and the rotational speed along the Z_c-axis we can deduce the translation speed along the Y_c -axis. The interaction matrix can be reformulated as follows:

$$L_s = \begin{bmatrix} -\frac{1}{z_d} & y_m \\ 0 & -x_m \end{bmatrix}. \tag{11}$$

Equation (9) can be rewritten in the following form:

$$\begin{pmatrix} \dot{x}_m \\ \dot{y}_m \end{pmatrix} = \begin{pmatrix} -\frac{1}{z_d} & y_m \\ 0 & -x_m \end{pmatrix} \begin{pmatrix} v_r \\ \omega_r \end{pmatrix}. \tag{12}$$

3.2.2. Creation of the Trajectory

It should be remembered that our objective is to carry out vehicle (target) tracking using a camera on board a quadrotor UAV. This problem can be converted into a problem of asymptotic tracking of a desired trajectory in the image plane of the point p resulting from the projection of the point P belonging to the vehicle. Let $(x^*(t), y^*(t))$ be this desired trajectory, as shown in Figure 2. Using Equation (12), the two mobile robot control inputs are given by

$$\begin{pmatrix} v_r \\ \omega_r \end{pmatrix} = \begin{pmatrix} -\frac{1}{z_d} & y_m \\ 0 & -x_m \end{pmatrix}^{-1} \begin{pmatrix} \dot{x_m} \\ \dot{y_m} \end{pmatrix}. \tag{13}$$

Assuming that point p does not coincide with the center of the projection (i.e., $x_m \neq 0$), and calculating the inverse of the interaction matrix, we get

$$\begin{pmatrix} v_r \\ \omega_r \end{pmatrix} = \begin{pmatrix} -z_d & -\frac{z_d y_m}{x_m} \\ 0 & -\frac{1}{x_m} \end{pmatrix} \begin{pmatrix} \dot{x}_m \\ \dot{y_m} \end{pmatrix}. \tag{14}$$

In this case, we have a reversible relationship between outputs and inputs. We use the exact linearization presented by Hagenmeyer–Delaleau in [38]. The resulting linearized system is equivalent to a system that has an integration of the following form:

$$\begin{cases} \dot{x}_m = \vartheta_x \\ \dot{y}_m = \vartheta_y \end{cases}, \tag{15}$$

where ϑ_x and ϑ_y are the two auxiliary control inputs to be specified, which ensure the asymptotic tracking of the desired trajectory. The control law of the mobile robot is finally given by

$$\begin{pmatrix} v_r \\ \omega_r \end{pmatrix} = \begin{pmatrix} -z_d & -\frac{z_d y_m}{x_m} \\ 0 & -\frac{1}{x_m} \end{pmatrix} \begin{pmatrix} \vartheta_x \\ \vartheta_y \end{pmatrix}, \tag{16}$$

with

$$\begin{cases} \vartheta_x = \dot{x}^* + k_1(x^* - x_m) \\ \vartheta_y = \dot{y}^* + k_2(y^* - y_m) \end{cases}. \tag{17}$$

k_1 and k_2 are chosen so that the error dynamics are asymptotically stable. In this case, it suffices to take $k_1 > 0$ and $k_2 > 0$, which ensure an asymptotic pursuit of the desired trajectory (x^*, y^*). The variables $(x^*, y^*, \dot{x}^*, \dot{y}^*)$ represent the metric coordinates in the image plane, in position and speed of the point P. The metric coordinates in the image plane of the desired point to be tracked all along the path are (x_m, y_m). Once the two commands (v_r, ω_r) ensuring the asymptotic tracking of the point P of the vehicle (target) are defined, we can deduce using Equation (6) the necessary movements (x_d, y_d, ψ_d) that the quadrotor must achieve to ensure the vehicle tracking.

3.3. Loop 3: Flatness-Based Tracking Control

In this subsection, we propose a control by flatness to achieve and ensure an asymptotic convergence of the trajectory generated in Section 3.2.

3.3.1. Flatness Theory

Flat systems theory is a complex area of research in differential algebra and differential geometry. Differential flatness is introduced as follows [29]:

A nonlinear system is given by:

$$\dot{x} = f(x, u),\ x \in \Re^n \text{ and } u \in \Re^m. \tag{18}$$

This system is differentially flat if there is a vector, $F \in \Re^m$, such that

$$F = \xi\left(x, u, \dot{u}, \ldots, u^{(r-1)}\right), \tag{19}$$

whose components are differentially independent, and two functions, $\eta(.)$ and $\Gamma(.)$, such that

$$x = \eta\left(F, \dot{F}, \ddot{F}, \ldots, F^{(\alpha-1)}\right) \tag{20}$$

$$u = \Gamma\left(F, \dot{F}, \ddot{F}, \ldots, F^{(\alpha)}\right), \tag{21}$$

where α and r and are finite multi-indices and ξ, η, and Γ are vectors of smooth functions. The vector F that appears in this definition is called the flat output of the system. In other words, a flat system is a system whose state and control variables can be written according to this flat output and its derivatives. The open-loop flatness control given by Equation (21) is known as the Brunovosky control because it provides an exact linearization of the system. For a differentially flat system, when the desired trajectory, F_d, is known, the desired state, x_d, and the desired open-loop control, u_d, can be defined as follows:

$$x_d = \eta\left(F_d, \dot{F}_d, \ddot{F}_d, \ldots, F_d^{(\alpha-1)}\right), \tag{22}$$

$$u_d = \Gamma\left(F_d, \dot{F}_d, \ddot{F}_d, \ldots, F_d^{(\alpha)}\right). \tag{23}$$

If the system is naturally stable, it will behave well and follow the desired trajectory. For unstable systems whose purpose is to accelerate convergence, it is necessary to add to this open-loop control a small closed-loop correction term to ensure trajectory tracking.

In this work, a closed-loop flatness control is proposed. We denote it by FTC: flatness-based tracking control. This control contains two parts: the open loop control given by

Equation (21) and a loop term, ϑ, which represents a linear control capable of stabilizing the obtained linearized system. The FTC is given as follows:

$$u_{FTC} = \Gamma\left(F_d, \dot{F}_d, \ddot{F}_d, \ldots, \vartheta\right). \tag{24}$$

$\vartheta(t)$ represents the new command. When $\frac{\partial \Gamma_{(.)}}{\partial F^{(\alpha)}}$ is locally invertible, this leads to the following decoupled system:

$$F^{(\alpha)} = \vartheta, \tag{25}$$

with

$$\vartheta = F_d^{(\alpha)} + \sum_{i=0}^{\alpha-1} k_i \left(F_d^{(i)} - F^{(i)}\right). \tag{26}$$

Let $K(p) = p^{\beta} + \sum_{i=0}^{\beta-1} k_i p^i$ be a diagonal matrix whose elements are polynomials with negative real part roots. This allows for asymptotic trajectory tracking with

$$\lim_{t\to\infty} (F_d - F) = 0. \tag{27}$$

3.3.2. Control Strategy

As shown in Figure 1, the control used to ensure the realization of the movements necessary for the quadrotor in order to satisfy the tracking of the vehicle (target) is based on the differential flatness. This control uses an open-loop control that linearizes the system and a closed-loop correction term that ensures the asymptotic convergence of the desired trajectory even in the presence of disturbances. Here, we replace the command u_1 of Equation (3) in the model that describes the dynamics of the quadrotor (Equation (1)):

$$\begin{cases} \ddot{x} = (Nu_z + g)tan\theta cos\psi + (Nu_z + g)\frac{\tan(\phi)}{cos\theta} sin\psi \\ \ddot{y} = (Nu_z + g)tan\theta sin\psi - (Nu_z + g)\frac{\tan(\phi)}{cos\theta} cos\psi \\ \ddot{z} = Nu_z = \ddot{z}_d + k_{11}(\dot{z}_d - \dot{z}) + k_{12}(z_d - z) \\ \ddot{\theta} = u_2 \\ \ddot{\phi} = u_3 \\ \ddot{\psi} = u_4 \end{cases}. \tag{28}$$

We prove that this system is flat and has flat output: $F_1 = z$; $F_2 = x$; $F_3 = y$; $F_4 = \psi$. Indeed, using the first and second lines of Equation (28), we can express the variables θ and ϕ in terms of the flat outputs:

$$\begin{cases} \theta = arctan\left(\frac{cos(\psi)\ddot{x} + sin(\psi)\ddot{y}}{\ddot{z}+g}\right) \\ \phi = arcsin\left(\frac{sin(\psi)\ddot{x} - cos(\psi)\ddot{y}}{\sqrt{\ddot{x}^2 + \ddot{y}^2 + (\ddot{z}+g)^2}}\right) \end{cases}. \tag{29}$$

Control expressions can be written in terms of flat outputs and their derivatives:

$$\begin{cases} u_2 = \ddot{\theta} = \frac{d^2}{dt}\left(arctan\left(\frac{cos(\psi)\ddot{x} + sin(\psi)\ddot{y}}{\ddot{z}+g}\right)\right) \\ u_3 = \ddot{\phi} = \frac{d^2}{dt}\left(arcsin\left(\frac{sin(\psi)\ddot{x} - cos(\psi)\ddot{y}}{\sqrt{\ddot{x}^2 + \ddot{y}^2 + (\ddot{z}+g)^2}}\right)\right) \\ u_4 = \ddot{\psi} \end{cases}. \tag{30}$$

We have just expressed all the variables of the system as a function of the dynamics of (z, x, y, ψ). The system of Equation (28) is flat and has as flat outputs of $F_1 = z$; $F_2 = x$; $F_3 = y$; $F_4 = \psi$. To achieve the desired trajectory (x_d, y_d, ψ_d), generated by the trajectory

generation block and using Equation (30), we can deduce the open loop control ensuring this desired trajectory:

$$\begin{cases} u_{2d} = \frac{d^2}{dt}\left(arctan\left(\frac{cos(\psi_d)\ddot{x}_d + cos(\psi_d)\ddot{y}_d}{\ddot{z}_d + g}\right)\right) \\ u_{3d} = \frac{d^2}{dt}\left(arcsin\left(\frac{sin(\psi_d)\ddot{x}_d - cos(\psi_d)\ddot{y}_d}{\sqrt{\ddot{x}_d^2 + \ddot{y}_d^2 + \left(\ddot{z}_d + g\right)^2}}\right)\right) \\ u_{4d} = \ddot{\psi}_d \end{cases} . \tag{31}$$

Until now, flatness has been used to calculate the commands corresponding to the open-loop trajectories of the system. If the system is intrinsically stable, it will behave appropriately and pursue the desired trajectory. For unstable systems whose purpose is to accelerate convergence, a small closed-loop correction term must be added to this open-loop control to guarantee trajectory tracking. To generate this correction term, we will consider some simplification assumptions. It should be noted here that these assumptions relate only to the development of a correction term that is considered in the vicinity of the desired trajectory.

When the quadrotor joins the desired trajectory, we can assume that the angles θ, ϕ, and ψ become small. The expressions for the second derivative of θ and ϕ will be given by

$$\begin{cases} \ddot{\theta} = \frac{F_2^{(4)}}{\ddot{F}_1 + g} - 2\frac{F_2^{(3)} F_1^{(3)}}{\left(\ddot{F}_1 + g\right)^2} + 2\frac{\ddot{F}_1\left(F_1^{(3)}\right)^2}{\left(\ddot{F}_1 + g\right)^3} - \frac{\ddot{F}_2 F_1^{(4)}}{\left(\ddot{F}_1 + g\right)^3} \\ \ddot{\phi} = \frac{F_3^{(4)}}{\ddot{F}_1 + g} + 2\frac{F_3^{(3)} F_1^{(3)}}{\left(\ddot{F}_1 + g\right)^2} - 2\frac{\ddot{F}_3\left(F_1^{(3)}\right)^2}{\left(\ddot{F}_1 + g\right)^3} + \frac{\ddot{F}_3 F_1^{(4)}}{\left(\ddot{F}_1 + g\right)^2} \end{cases} . \tag{32}$$

By using the theorem given in [39], which neglects all terms of the polynomial equation greater than the fourth degree, Equation (32) becomes

$$\begin{cases} \ddot{\theta} = \frac{F_2^{(4)}}{\ddot{F}_1 + g} \\ \ddot{\phi} = \frac{F_3^{(4)}}{\ddot{F}_1 + g} \end{cases} . \tag{33}$$

Assuming the quadrotor reaches its desired altitude ($z - z_d = 0$), the control expressions are then

$$\begin{cases} u_2 = \ddot{\theta} = \frac{F_2^{(4)}}{g} \\ u_3 = \ddot{\phi} = \frac{F_3^{(4)}}{g} \end{cases} . \tag{34}$$

The expressions of the closed-loop control laws (FCT), which ensure asymptotic convergence toward the desired trajectory even in the presence of disturbances, are given by

$$\begin{cases} u_{2FTC} = u_{2d} + k_{21} e_2^{(3)} + k_{22} e_2^{(2)} + k_{23}\dot{e}_2 + k_{24} e_2 \\ u_{3FTC} = u_{3d} + k_{31} e_3^{(3)} + k_{32} e_3^{(2)} + k_{33}\dot{e}_3 + k_{34} e_3 \\ u_{4FTC} = u_{4d} + k_{41}\dot{e}_4 + k_{42} e_4 \end{cases} \tag{35}$$

with $e_i = F_{id} - F_i$ $(i = 2,\ 3,\ 4)$, and the k_{ij} values are deduced using the pole placement technique.

4. Simulation Results

To demonstrate the effectiveness of our proposed control strategy, we divide this section into two parts. In the first part, we exclusively test the algorithm responsible for generating the necessary movements of the quadrotor based on its dynamics, which are similar to those of a differential robot. It is essential to assess its ability to maintain the target in the center of the image plane. Then, in the second part, we integrate this algorithm with the other control loops using the complete model of the quadrotor. The tool used to perform the simulations is the RVCTOOLS library from MATLAB 9.2 R2017b.

4.1. Proposed Algorithm Performance Related to the FOV Constraint

Since we are in a simulated environment, in order to generate the displacements necessary for the quadrotor to ensure the pursuit of a vehicle (target), we must propose a model for this vehicle. This model will be used to generate the trajectory of a point in the image plane with variable dynamics. We seek to prove that the proposed approach generates a trajectory for the quadrotor that fully copies the dynamics of the vehicle (target) without a priori knowledge of this dynamic. The only knowledge available is that of the instantaneous position (as well as the history of this position) in pixels of the point p of the vehicle (target). This vehicle is a car-type vehicle, as is described by Figure 4. We assume that this vehicle is located just below the quadrotor and that it starts to move according to the following kinematic model:

$$\begin{cases} \dot{x}_v = v_v cos(\theta_v) \\ \dot{y}_v = v_v sin(\theta_v) \\ \dot{\theta}_v = \omega_v \end{cases} \tag{36}$$

where v_v is the linear speed of the vehicle given by $v_r = \frac{r}{4}(\omega_1 + \omega_2)$ and ω_v is the angular speed of the vehicle given by $\omega_v = \frac{r}{2R}(\omega_2 - \omega_1)$.

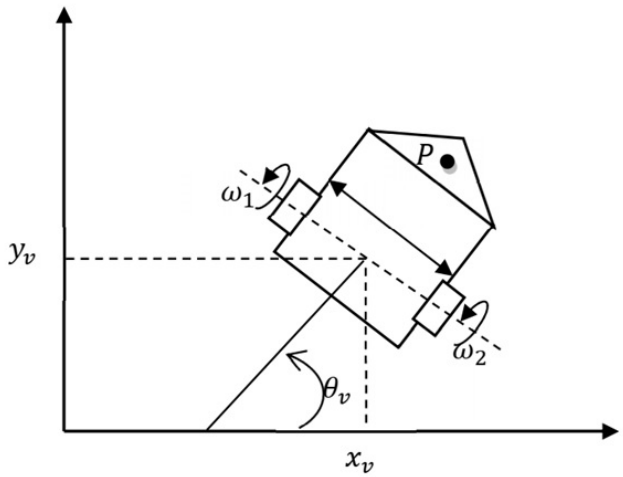


Figure 4. Considered car-type vehicle (target).

The position of the vehicle is given by $X_v = [x_v, y_v, \theta_v]^T$. The two control inputs are v_v and ω_v. (x_v, y_v) represent the abscissa and the ordinate of the middle of the axis of the two driving wheels. θ_v represents the orientation of the vehicle. ω_1 and ω_2 are the speeds of the two driving wheels. R is the distance between the two wheels, and r is the diameter of a wheel. The movement of the vehicle is managed by the two rotational speeds (ω_1, ω_2) of the two driving wheels. In order to achieve any movement at variable speed and orientation and to ensure a variation of the linear velocities of translation and the angular velocities of rotation, we have chosen (ω_1, ω_2), as follows:

$$\begin{cases} \omega_1(t) = 20\sin(0.2\pi t) \\ \omega_2(t) = 20\sin(0.22\pi t) \end{cases}, \text{for } t \in [0,\ 120\text{s}]. \tag{37}$$

In order to facilitate the detection of the descriptor point of the target object (the vehicle), we have chosen the center of gravity of the vehicle's hood. Our objective is to keep the vehicle (target) centered in the image plane of the quadrotor camera throughout its movement. So that the coordinates of this point are not largely affected by displacements along the Z-axis, it is better that this point is on the horizontal axis, which passes through the center of projection, and that it is not confused with the throwing center. Indeed, the center of projection is invariant with respect to the rotation along the Z-axis. The desired image that must be satisfied throughout the movement of the vehicle is given in Figure 5.

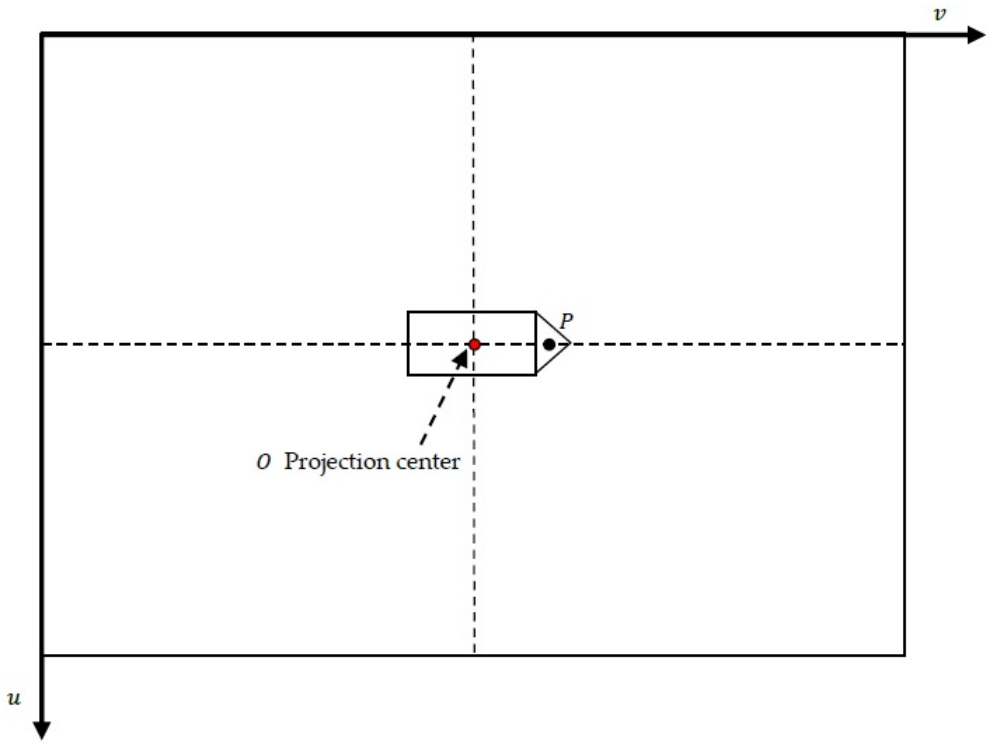
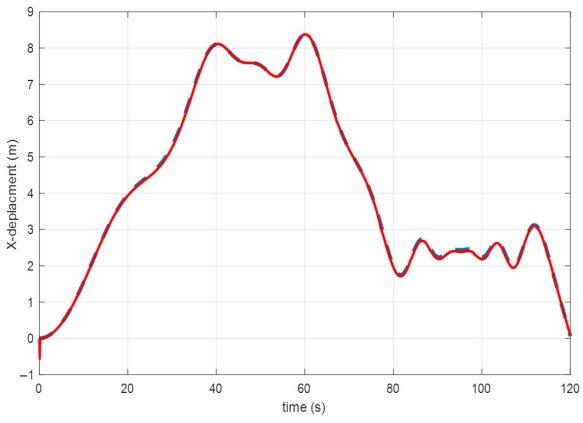
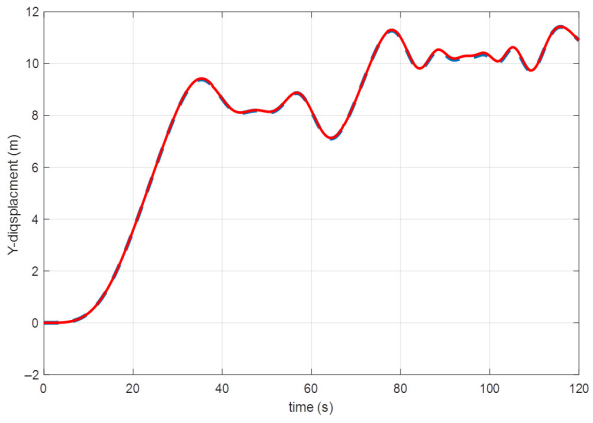


Figure 5. Desired image.

The simulation results are given in Figure 6. Figure 6a–d illustrate the displacement along the *X*- and *Y*-axes as well as the orientation along the *Z*-axis, both real and generated by the visual servo loop. Figure 6e–g show linear translational velocities and angular rotational velocities, respectively. It is clear that despite variations in vehicle speed, the desired trajectory is followed perfectly. Figure 6h shows the continuous detection of point *P* in the image plane along the trajectory. It is obvious that the object always remains in the field of view of the camera. In a practical situation, it would be possible to adjust the altitude of the quadrotor to widen the field of view because our method allows this variable (the altitude) to be independently controlled from the others.

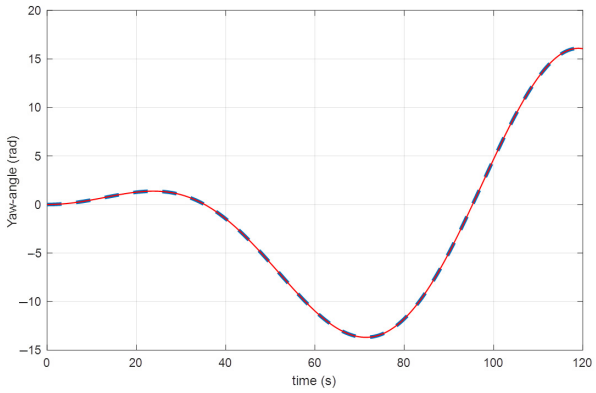


(**a**)

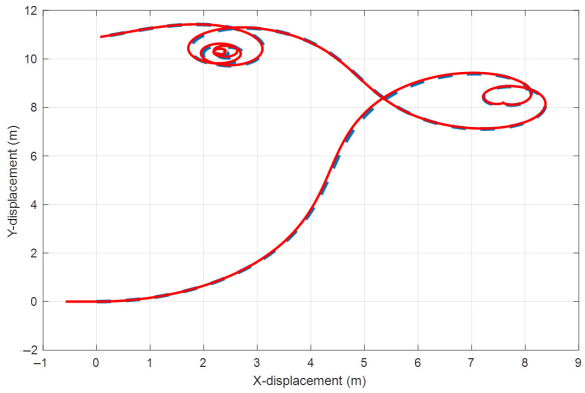


(**b**)

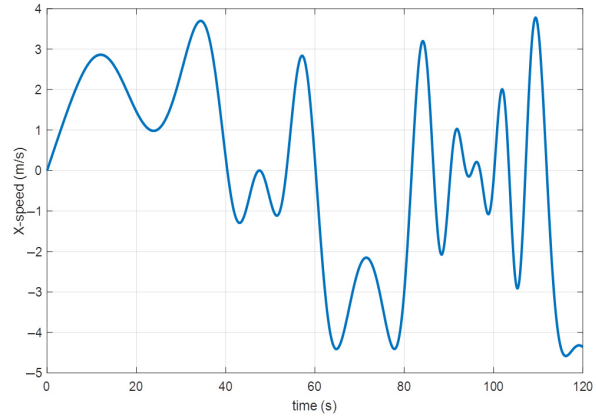
Figure 6. *Cont.*



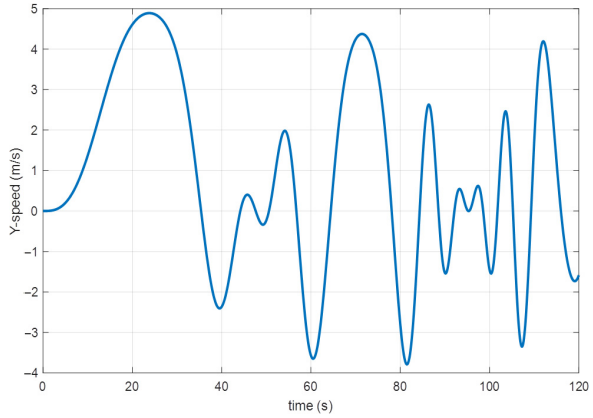
(**c**)



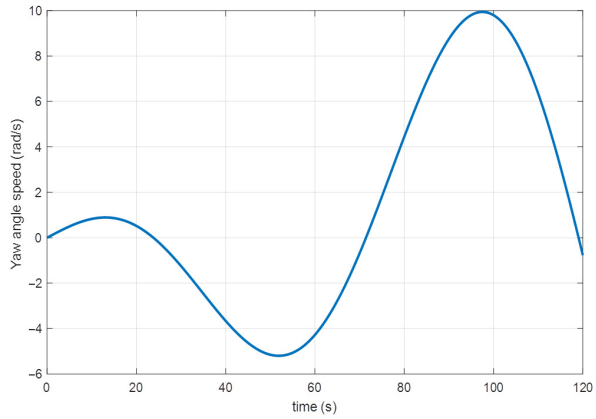
(**d**)



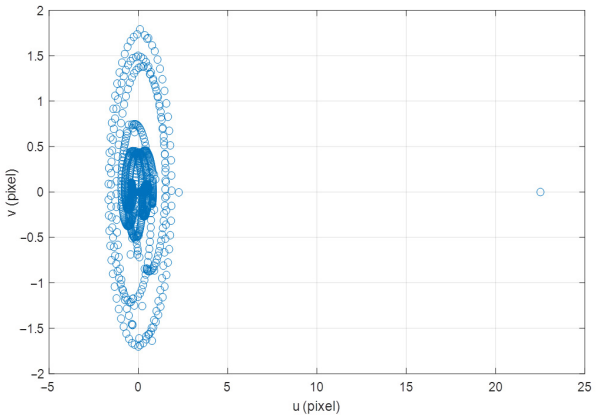
(**e**)



(**f**)



(**g**)



(**h**)

Figure 6. Performances related to the FOV constraint. (**a**–**d**) Displacement along the *X*- and *Y*-axes as well as the orientation along the *Z*-axis, both real (in red) and generated by the visual servo loop (in blue). (**e**–**g**) Linear translational velocities and angular rotational velocities. (**h**) Continuous detection of point *P* (target point) in the image plane.

4.2. Global Tracking Strategy's Performance

To validate the effectiveness of the global tracking strategy, we will consider the following experiment. The quadrotor takes off to reach a given altitude. The vehicle (target) is located just below the quadrotor. At a given moment, it begins to move according to a variable dynamic. As mentioned above, to ensure the pursuit or tracking of the vehicle, we have made three servo loops. In this section, we detail the simulation results of each loop.

We include in this experiment the fact that the quadrotor cannot maintain a constant altitude throughout its flight. We choose the descriptor point P such that its projection on the image plane is located on the horizontal axis H which passes through the center of projection. This point will be less affected by the displacement along the Z-axis. We impose a variable altitude given by the following expression: $z_d(t) = 0.1sin(0.04\pi t) + 5$ for $t \in [0,\ 200\text{s}]$. The control law responsible for altitude control is given by Equations (3) and (5). The parameters of Equation (5) are chosen following experimental tests: $k_{11} = 10$; $k_{12} = 25$. The simulation results are given in Figure 7.

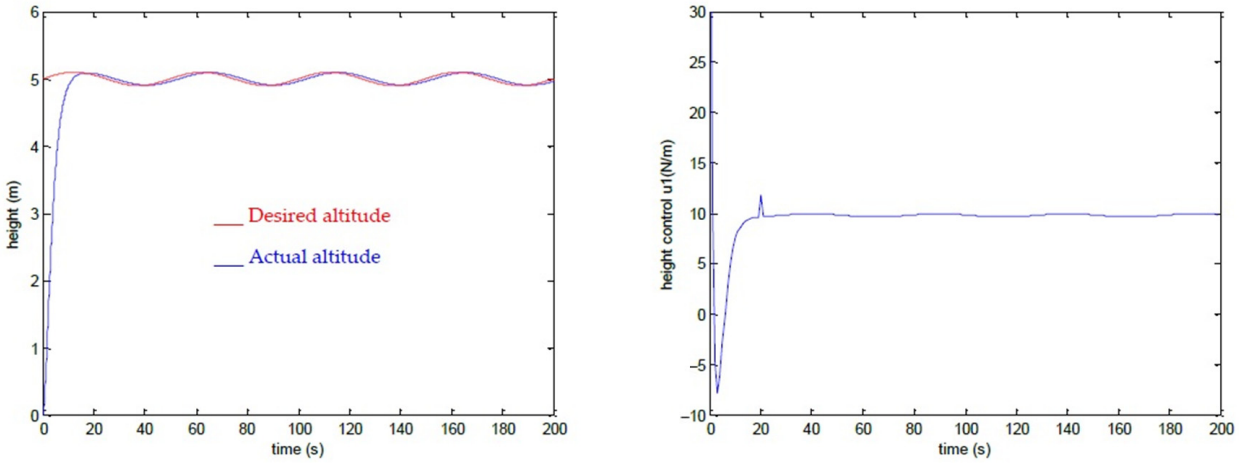


Figure 7. Loop1 simulation results. (**a**) Desired altitude vs. actual altitude. (**b**) Altitude control variations u_1.

Figure 7a represents the desired altitude and the altitude achieved by the quadrotor. We can clearly see that the quadrotor ensures the tracking of this altitude. Figure 7b represents the total force that is responsible for the displacement along the Z-axis. We note that it is a continuous command and physically realizable. In order to achieve any movement at variable speed and orientation, we have chosen (ω_1,ω_2) as follows:

$$\begin{cases} \omega_1(t) = 20\sin(0.00026\pi t) + 20 \\ \omega_2(t) = 20\sin(0.0002\pi t) + 20 \end{cases}, \text{ for } t \in [20,\ 200\text{s}]. \tag{38}$$

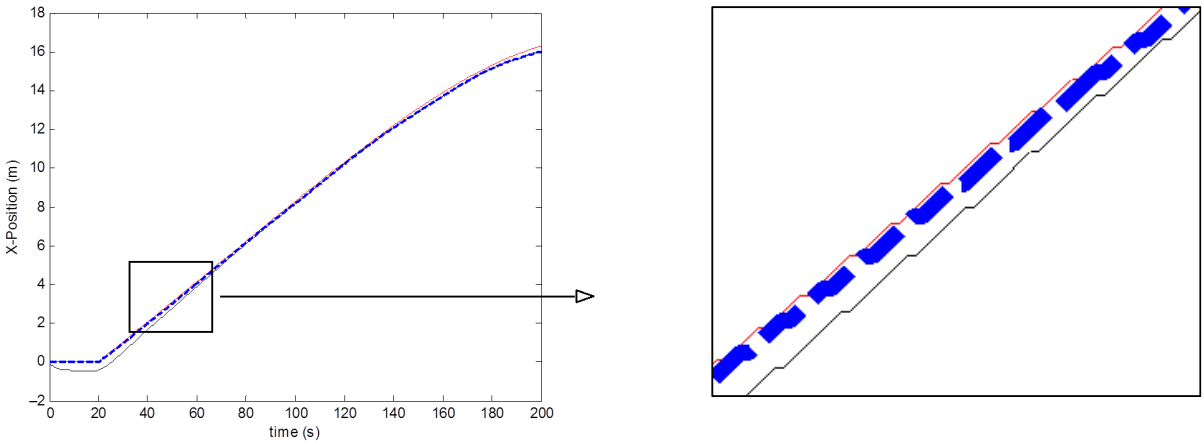
The 2D dynamic visual servoing is provided by Equations (16) and (17). The coefficients of Equation (17) are given by $k_1 = 28$ and $k_2 = 28$. To realize the trajectory thus generated, a flatness-based control is proposed. The values of the gain that control the dynamics of the errors are given in Table 1:

Table 1. Gain values associated with the dynamics of the errors.

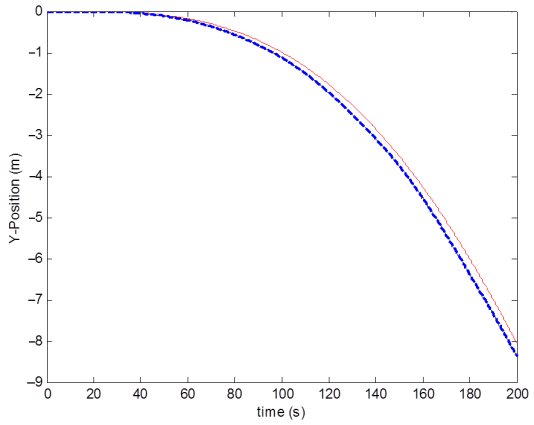
Gain	k_{21}	k_{22}	k_{23}	k_{24}	k_{31}	k_{32}	k_{33}	k_{34}	k_{41}	k_{42}
Value	4	6	4	1	4	6	4	1	2	1

The simulation results are given by Figure 8. Figure 8a–c represent, respectively, the actual trajectory (displacement along X, displacement along Y, and orientation along the Z-axis) performed by the vehicle, the trajectory generated by the visual servo loop and the trajectory produced by the quadrotor using flatness control. We notice that the visual servoing algorithm generates a trajectory faithful to the real trajectory carried out by the vehicle (target). Flatness control ensures exact tracking of the generated trajectory. The evolution of the error in pixels between the position of the point p in the image plane and the desired point is given in Figure 8d,e. The evolution of the roll angle and the pitch angle,

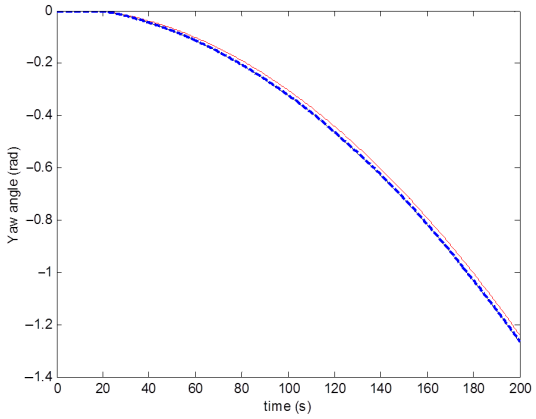
respectively, are given in Figure 8f,g. We notice that these two variables remain sufficiently small during the trajectory tracking.



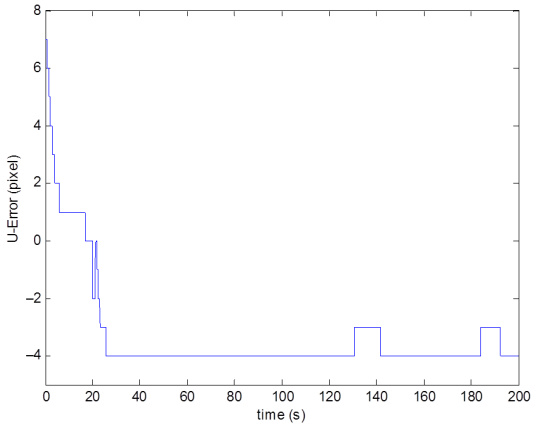
(**a**)



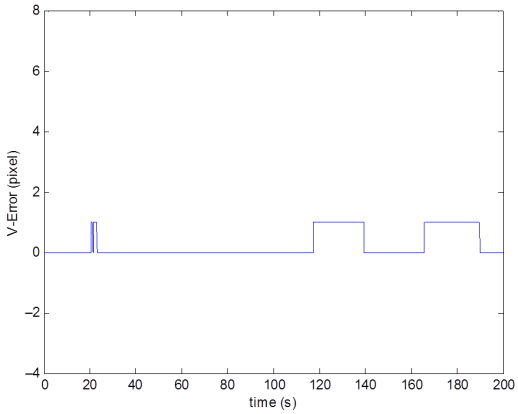
(**b**)



(**c**)

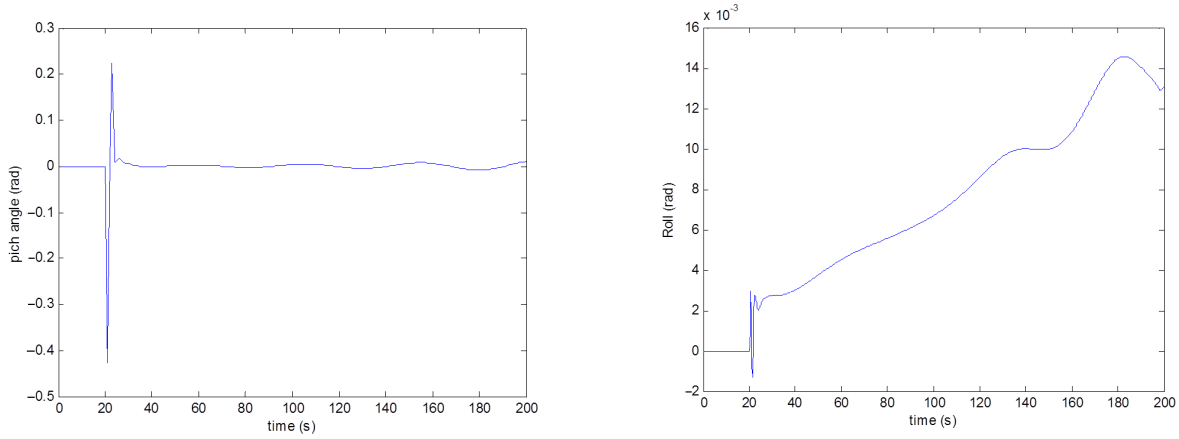


(**d**)



(**e**)

Figure 8. *Cont.*



(**f**) (**g**)

Figure 8. Simulation results. (**a**) Displacement along X of the target trajectory (red), the visual servo-based trajectory (gray), and the flatness-based quadrotor trajectory (blue). (**b**) Displacement along Y of the target trajectory (red), the visual servo-based trajectory (gray), and the flatness-based quadrotor trajectory (blue). (**c**) Orientation along Z (Yaw angle) of the target trajectory (red), the visual servo-based trajectory (gray), and the flatness-based quadrotor trajectory (blue). (**d**) Error evolution in pixels along u. (**e**) Error evolution in pixels along v. (**f**) The evolution of the roll angle. (**g**) The evolution of the pitch angle.

5. Conclusions

In this paper, we have presented a new dynamic image-based visual servoing method. We proposed to solve the problem of the pursuit of a car-type vehicle by a quadrotor UAV. Under the specific conditions of this application, we have demonstrated that we can use a single point of the target object to perform the task of dynamic visual servoing. This contribution aims to reduce the computation time of the quadrotor control law. To circumvent the problem of controlling an under-actuated system and to achieve the necessary displacements generated by the visual servoing algorithm, a new flatness-based control algorithm has been integrated. The simulation results show the effectiveness of the proposed method. The proposed method provides an intriguing solution to the issue of vehicle tracking by a quadrotor UAV through the utilization of an onboard camera. In order to effectively implement this method, it is advisable to pair it with an additional algorithm that enables the precise selection of the target vehicle from among the other automobiles present. Given the aforementioned considerations, we suggest a classification algorithm based on artificial intelligence. In another context, the proposed method can be extended to solve the planning problem in the image plane for the 2D visual servoing of a quadrotor. This represents a challenge and will solve many problems related to the integration of vision in UAVs.

Author Contributions: Conceptualization, H.M. and K.K.; methodology, H.M. and A.A.; software, A.A., B.A.A. and S.A.; validation, H.M., A.A. and K.B.; formal analysis, K.B. and K.K.; investigation, H.M. and M.A.; resources, A.S. and M.A.; writing—original draft preparation, H.M. and K.K.; writing—review and editing, A.A. and A.S.; visualization, A.A. and S.A.; supervision, H.M.; project administration, K.K.; funding acquisition, A.A. All authors have read and agreed to the published version of the manuscript.

Funding: The authors extend their appreciation to the Deputyship for Research and Innovation, Ministry of Education in Saudi Arabia for funding this research work through the project number 223202.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Wei, Z.; Zhu, M.; Zhang, N.; Wang, L.; Zou, Y.; Meng, Z.; Wu, H.; Feng, Z. UAV-Assisted Data Collection for Internet of Things: A Survey. *IEEE Internet Things J.* **2022**, *9*, 15460–15483. [CrossRef]
2. Ceren, Z.; Altuğ, E. Image Based and Hybrid Visual Servo Control of an Unmanned Aerial Vehicle. *J. Intell. Robot. Syst.* **2011**, *65*, 325–344. [CrossRef]
3. Metni, N.; Hamel, T. A UAV for Bridge Inspection: Visual Servoing Control Law with Orientation Limits. *Autom. Constr.* **2007**, *17*, 3–10. [CrossRef]
4. Arafat, M.Y.; Moh, S. Routing Protocols for Unmanned Aerial Vehicle Networks: A Survey. *IEEE Access* **2019**, *7*, 99694–99720. [CrossRef]
5. Fraundorfer, F.; Heng, L.; Honegger, D.; Lee, G.H.; Meier, L.; Tanskanen, P.; Pollefeys, M. Vision-Based Autonomous Mapping and Exploration Using a Quadrotor MAV. In Proceedings of the 2012 IEEE/RSJ International Conference on Intelligent Robots and Systems, Algarve, Portugal, 7–12 October 2012. [CrossRef]
6. Bi, R.; Gan, S.; Yuan, X.; Li, R.; Gao, S.; Yang, M.; Luo, W.; Hu, L. Multi-View Analysis of High-Resolution Geomorphic Features in Complex Mountains Based on UAV–LiDAR and SfM–MVS: A Case Study of the Northern Pit Rim Structure of the Mountains of Lufeng, China. *Appl. Sci.* **2023**, *13*, 738. [CrossRef]
7. Tokekar, P.; Vander Hook, J.; Mulla, D.; Isler, V. Sensor Planning for a Symbiotic UAV and UGV System for Precision Agriculture. In Proceedings of the 2013 IEEE/RSJ International Conference on Intelligent Robots and Systems, Tokyo, Japan, 3–7 November 2013. [CrossRef]
8. Costello, B.; Osunkoya, O.O.; Sandino, J.; Marinic, W.; Trotter, P.; Shi, B.; Gonzalez, F.; Dhileepan, K. Detection of Parthenium Weed (*Parthenium hysterophorus* L.) and Its Growth Stages Using Artificial Intelligence. *Agriculture* **2022**, *12*, 1838. [CrossRef]
9. Chaumette, F.; Hutchinson, S. Visual Servo Control. I. Basic Approaches. *IEEE Robot. Autom. Mag.* **2006**, *13*, 82–90. [CrossRef]
10. Keshmiri, M.; Xie, W.-F. Image-Based Visual Servoing Using an Optimized Trajectory Planning Technique. *IEEE/ASME Trans. Mechatron.* **2017**, *22*, 359–370. [CrossRef]
11. Allibert, G.; Courtial, E.; Chaumette, F. Predictive Control for Constrained Image-Based Visual Servoing. *IEEE Trans. Robot.* **2010**, *26*, 933–939. [CrossRef]
12. Qiu, Z.; Hu, S.; Liang, X. Model Predictive Control for Uncalibrated and Constrained Image-Based Visual Servoing without Joint Velocity Measurements. *IEEE Access* **2019**, *7*, 73540–73554. [CrossRef]
13. Gao, J.; Proctor, A.A.; Shi, Y.; Bradley, C. Hierarchical Model Predictive Image-Based Visual Servoing of Underwater Vehicles With Adaptive Neural Network Dynamic Control. *IEEE Trans. Cybern.* **2016**, *46*, 2323–2334. [CrossRef]
14. Chen, C.-W.; Hung, H.-A.; Yang, P.-H.; Cheng, T.-H. Visual Servoing of a Moving Target by an Unmanned Aerial Vehicle. *Sensors* **2021**, *21*, 5708. [CrossRef]
15. Heshmati-alamdari, S.; Karras, G.C.; Eqtami, A.; Kyriakopoulos, K.J. A Robust Self Triggered Image Based Visual Servoing Model Predictive Control Scheme for Small Autonomous Robots. In Proceedings of the 2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Hamburg, Germany, 28 September–3 October 2015. [CrossRef]
16. Razzanelli, M.; Innocenti, M.; Pannocchia, G.; Pollini, L. Vision-Based Model Predictive Control for Unmanned Aerial Vehicles Automatic Trajectory Generation and Tracking. In Proceedings of the AIAA Scitech 2019 Forum, San Diego, CA, USA, 7–11 January 2019. [CrossRef]
17. Shi, H.; Sun, G.; Wang, Y.; Hwang, K.-S. Adaptive Image-Based Visual Servoing With Temporary Loss of the Visual Signal. *IEEE Trans. Ind. Inform.* **2019**, *15*, 1956–1965. [CrossRef]
18. Wang, H.; Yang, B.; Wang, J.; Liang, X.; Chen, W.; Liu, Y.-H. Adaptive Visual Servoing of Contour Features. *IEEE/ASME Trans. Mechatron.* **2018**, *23*, 811–822. [CrossRef]
19. Pan, W.; Lyu, M.; Hwang, K.-S.; Ju, M.-Y.; Shi, H. A Neuro-Fuzzy Visual Servoing Controller for an Articulated Manipulator. *IEEE Access* **2018**, *6*, 3346–3357. [CrossRef]
20. Pence, W.G.; Farelo, F.; Alqasemi, R.; Sun, Y.; Dubey, R. Visual Servoing Control of a 9-DoF WMRA to Perform ADL Tasks. In Proceedings of the 2012 IEEE International Conference on Robotics and Automation, Saint Paul, MN, USA, 14–18 May 2012. [CrossRef]
21. La Anh, T.; Song, J.-B. Robotic Grasping Based on Efficient Tracking and Visual Servoing Using Local Feature Descriptors. *Int. J. Precis. Eng. Manuf.* **2012**, *13*, 387–393. [CrossRef]
22. Penin, B.; Giordano, P.R.; Chaumette, F. Vision-Based Reactive Planning for Aggressive Target Tracking While Avoiding Collisions and Occlusions. *IEEE Robot. Autom. Lett.* **2018**, *3*, 3725–3732. [CrossRef]
23. Siradjuddin, I.; Tundung, S.P.; Indah, A.S.; Adhisuwignjo, S. A Real-Time Model Based Visual Servoing Application for a Differential Drive Mobile Robot Using Beaglebone Black Embedded System. In Proceedings of the 2015 IEEE International Symposium on Robotics and Intelligent Sensors (IRIS), Langkawi, Malaysia, 18–20 October 2015. [CrossRef]
24. Hulens, D.; Goedeme, T. Autonomous Flying Cameraman with Embedded Person Detection and Tracking While Applying Cinematographic Rules. In Proceedings of the 2017 14th Conference on Computer and Robot Vision (CRV), Edmonton, AB, Canada, 17–19 May 2017. [CrossRef]
25. Liu, Y.; Wang, Q.; Hu, H.; He, Y. A Novel Real-Time Moving Target Tracking and Path Planning System for a Quadrotor UAV in Unknown Unstructured Outdoor Scenes. *IEEE Trans. Syst. Man Cybern. Syst.* **2019**, *49*, 2362–2372. [CrossRef]

26. Cao, Z.; Chen, X.; Yu, Y.; Yu, J.; Liu, X.; Zhou, C.; Tan, M. Image Dynamics-Based Visual Servoing for Quadrotors Tracking a Target With a Nonlinear Trajectory Observer. *IEEE Trans. Syst. Man Cybern. Syst.* **2020**, *50*, 376–384. [CrossRef]
27. Limon, D.; Pereira, M.; Munoz de la Pena, D.; Alamo, T.; Jones, C.N.; Zeilinger, M.N. MPC for Tracking Periodic References. *IEEE Trans. Autom. Control* **2016**, *61*, 1123–1128. [CrossRef]
28. López, J.; Dormido, R.; Dormido, S.; Gómez, J.P. A RobustH∞Controller for an UAV Flight Control System. *Sci. World J.* **2015**, *2015*, 1–11. [CrossRef] [PubMed]
29. Fliess, M.; Lévine, J.; Martin, P.; Rouchon, P. On Differentially Flat Nonlinear Systems. *IFAC Proc. Vol.* **1992**, *25*, 159–163. [CrossRef]
30. Zheng, D.; Wang, H.; Chen, W.; Wang, Y. Planning and Tracking in Image Space for Image-Based Visual Servoing of a Quadrotor. *IEEE Trans. Ind. Electron.* **2018**, *65*, 3376–3385. [CrossRef]
31. Zheng, D.; Wang, H.; Wang, J.; Chen, S.; Chen, W.; Liang, X. Image-Based Visual Servoing of a Quadrotor Using Virtual Camera Approach. *IEEE/ASME Trans. Mechatron.* **2017**, *22*, 972–982. [CrossRef]
32. Asl, H.J.; Oriolo, G.; Bolandi, H. An adaptive scheme for image-based visual servoing of an underactuated UAV. *Int. J. Robot. Autom.* **2014**, *29*, 92–104. [CrossRef]
33. Gomez-Avila, J.; Lopez-Franco, C.; Alanis, A.Y.; Arana-Daniel, N.; Lopez-Franco, M. Ground Vehicle Tracking with a Quadrotor Using Image Based Visual Servoing. *IFAC-PapersOnLine* **2018**, *51*, 344–349. [CrossRef]
34. Surma, C.C.; Barczyk, M. Linear Model Predictive Control for Vision-Based UAV Pursuit. *J. Unmanned Veh. Syst.* **2020**, *8*, 334–363. [CrossRef]
35. Chamseddine, A.; Li, T.; Zhang, Y.; Rabbath, C.A.; Theilliol, D. Flatness-Based Trajectory Planning for a Quadrotor Unmanned Aerial Vehicle Test-Bed Considering Actuator and System Constraints. In Proceedings of the 2012 American Control Conference (ACC), Montreal, QC, Canada, 27–29 June 2012. [CrossRef]
36. Li, T.; Xia, Y.; Ma, D. Flatness-Based Target Tracking for a Quadrotor Unmanned Aerial Vehicle. *IFAC-PapersOnLine* **2015**, *48*, 874–879. [CrossRef]
37. Abadi, A.; El Amraoui, A.; Mekki, H.; Ramdani, N. Guaranteed Trajectory Tracking Control Based on Interval Observer for Quadrotors. *Int. J. Control* **2019**, *93*, 2743–2759. [CrossRef]
38. Hagenmeyer, V.; Delaleau, E. Exact Feedforward Linearization Based on Differential Flatness. *Int. J. Control* **2003**, *76*, 537–556. [CrossRef]
39. Chamseddine, A.; Zhang, Y.; Rabbath, C.A.; Theilliol, D. Trajectory Planning and Replanning Strategies Applied to a Quadrotor Unmanned Aerial Vehicle. *J. Guid. Control Dyn.* **2012**, *35*, 1667–1671. [CrossRef]