drones

MDPI

Article

Sliding Surface Designs for Visual Servo Control of Quadrotors

Tolga Yuksel

Department of Electrical and Electronics Engineering, Engineering Faculty, Bilecik Seyh Edebali University, Bilecik 11200, Turkey; tolga.yuksel@bilecik.edu.tr

Abstract: Autonomy is the main task of a quadrotor, and visual servoing assists with this task while providing fault tolerance under GPS failure. The main approach to visual servoing is image-based visual servoing, which uses image features directly without the need for pose estimation. The classical sliding surface design of sliding mode control is used by the linear controller law of image-based visual servoing, and focuses only on minimizing the error in the image features as convergence. In addition to providing convergence, performance characteristics such as visual-feature-convergence time, error, and motion characteristics should be taken into consideration while controlling a quadrotor under velocity limitations and disturbance. In this study, an image-based visual servoing system for quadrotors with five different sliding surface designs is proposed using analytical techniques and fuzzy logic. The proposed visual servo system was simulated, utilizing the moment characteristics of a preset shape to demonstrate the effectiveness of these designs. The stated parameters, convergence time, errors, motion characteristics, and length of the path, followed by the quadrotor, were compared for each of these design approaches, and a convergence time that was 46.77% shorter and path length that was 6.15% shorter were obtained by these designs. In addition to demonstrating the superiority of the designs, this study can be considered as a reflection of the realization, as well as the velocity constraints and disturbance resilience in the simulations.

Keywords: image-based visual servoing; sliding-surface designs; fuzzy logic

Citation: Yuksel, T. Sliding Surface Designs for Visual Servo Control of Quadrotors. *Drones* **2023**, *7*, 531. https://doi.org/10.3390/drones7080531

Academic Editor: Panagiotis Partsinevelos

Received: 27 June 2023
Revised: 27 July 2023
Accepted: 31 July 2023
Published: 14 August 2023

1. Introduction

Quadrotors are at the top of the UAV list, and autonomy is their primary task. Visual servoing (VS) has shown significant success in assisting with autonomy while providing fault tolerance under GPS failure [1]. It requires features such as points, lines, shapes, or segments from the current image, and extraction mechanisms to obtain them [2]. After this extraction, the characteristics of these features, such as the coordinates in the image plane or the area of the contour, are presented by vector $\boldsymbol{s}$. The difference between the requested feature vector $\boldsymbol{s}^*$ and $\boldsymbol{s}$ is assumed to be the error vector, and this vector is deployed in the VS control law to obtain the control signals of the system in terms of linear and angular velocities. Image-based visual servoing (IBVS), the main VS approach, utilizes $\boldsymbol{s}$ gathered directly from the image, does not require pose estimation, and shows robustness against errors in depth prediction. The use of IBVS is more appealing in practical applications thanks to these benefits, and in this study, it serves as the VS method. Furthermore, as the camera configuration, the eye-in-hand configuration is assumed as the camera is attached to the center of the quadrotor.

The majority of IBVS studies focus on various feature-extraction techniques [3], camera geometry approaches and classes [4], hybrid VS approaches [5], or solutions to problems in IBVS, such as Jacobian matrix singularity or a missing field of view (FOV) [6,7], and the linear controller approach of VS is assumed to be the adequate controller, as the errors in the feature vectors exponentially decline. In addition, other performance indicators, such as the feature-convergence time, the velocity limits of the platform, the length of the path followed by the UAV, and the computational costs, are real-time metrics allowing for UAV applications to decrease their operation times while increasing their accuracy. The first

use of VS for applications in UAVs was reported in [8]. A helicopter was assumed as an underactuated rigid body, and through the use of point features in the spherical image plane, the feature dynamics were included in the dynamic equations of the motion of this body. Nevertheless, the image plane disturbances were ignored. In a different study, the authors expanded this approach to include quadrotors with linear VS features [9]. Again, feature noise was disregarded as the authors only analyzed the noise in the quadrotor's attitude and torque signals. Although the use of spherical camera projection for VS is fairly common, the approach to quadrotor visual servoing in quadrotor control with spherical camera projection demonstrated that the issue of inappropriate maneuver characteristics in the Cartesian plane occurs with this configuration [10,11]. A quadrotor-stabilization system using an output-tracking-control approach with backstepping control and two-camera pose estimation was proposed in [12]. A VS framework employing features acquired from a unique spherical image projection was also presented [13]. Metni and Hamel focused on visual tracking for helicopters, and they proposed a system that makes use of the image data, an adaptive nonlinear-tracking-control law, and the homography matrix acquired using backstepping techniques [14]. The image noise was taken into account, but the authors did not provide the control, velocity signals, or velocities of the helicopter. Furthermore, the backstepping method was also used for autonomous landing on a moving ship [15] and to track a moving target [16]. A VS feedback controller based on homography was proposed in [17]. To confirm the desired equilibrium point and ensure that predetermined feature points remain inside the camera's FOV throughout the system's trajectory, they recreated attitude- and depth-ratio data. To present hover and landing control for a VTOL UAV on a moving platform, Hérissé et al. applied optical flow [18]. With the homography matrix generated from the gyro and the target's camera measurements, Plinval et al. suggested feedback laws for UAV stabilization [19]. To keep the target in the FOV throughout the task, Asl et al. devised an IBVS strategy for regulating the translational and yaw rotational movements of a quadrotor, which ensured robustness under uncertain conditions [20]. Thomas et al. proposed a virtual image plane, examined a cylindrical form in this plane, and used the obtained properties to propose a VS system that perched autonomously with an IBVS control law, considering quadrotor dynamics [21]. A VS control system based on nonlinear observers to estimate translational velocity and spherical features was proposed by Mebarki et al. for moving nonplanar point targets [22]. For the purpose of calculating the adaptive gain in the VS of a quadrotor to track moving targets, reinforcement learning, a popular deep-learning method, was also utilized [23]. The main objectives of each of these VS approaches are UAV stabilization, hovering, or autolanding, with visual feedback.

The sliding surface design of the conventional sliding mode control (SMC) is used by the linear controller law of IBVS, as detailed in the following section. A common approach to assigning an acceptable slope in conventional SMC is to define a slope that is neither overly large nor excessively small to avoid exceeding control restrictions, and vice versa, to define a slope that is both sufficiently large to reach the sliding surface and slide on this surface quickly. With IBVS, this involves choosing a gain that is sufficiently large to accelerate the convergence and sufficiently modest to avoid exceeding velocity constraints. Parsapour et al. proposed a kernel-based VS with a PI-type sliding mode to eliminate the tracking errors of a robot manipulator [24]. Liu et al. utilized a super-twisting SMC for a rivet-in-hole insertion system to provide continuous control inputs, exponentially decreasing the feature errors and the robustness against the manipulator errors and external disturbances [25]. Miranda-Moya et al. designed an adaptive SMC scheme for robustness against external perturbations while tracking image feature targets, but image noise and chattering in control signals were neglected [26].

This study proposed an IBVS system with five modified sliding surface designs for quadrotors using analytical techniques and fuzzy logic (FL). The basis of this study is the design approaches for the VS control of robot manipulators [27]. The first design made use of an FL-based linearly changing sliding surface. The impacts of gain on IBVS experienced by trials were utilized to construct the linguistic rules of FL. The inputs of the

FL unit were the error and error derivative norms to model the error dynamics. The second design altered the linear sliding surface by utilizing the sliding surface design with fixed acceleration, providing time optimality. In the third design, a nonlinear sliding surface was assigned using a tangent hyperbolic function with a width parameter function. The fourth design added an integral term with fuzzy logic to increase convergence performance. The last design utilized a nonlinear time-varying function to design a nonlinear sliding surface. The VS system is simulated utilizing the moment characteristics of a preset shape to demonstrate the effectiveness of these designs. The stated parameters, convergence time, errors, and motion characteristics were compared for each of these design approaches. The velocity restrictions and disturbance resilience in the simulations can be considered mirrors of realization even though this study demonstrates the superiority of the designs.

2. The Proposed IBVS System with Sliding Surface Designs

VS, and IBVS, aim to minimize the errors of feature vector $\boldsymbol{s}$ derived from the $\boldsymbol{k}$ features

$$\boldsymbol{e}(\boldsymbol{t}) = \boldsymbol{s}(\boldsymbol{t}) - \boldsymbol{s}^* \tag{1}$$

where $\boldsymbol{e}(\boldsymbol{t}) \in \mathbb{R}^k$ is the error vector and $\boldsymbol{s}(\boldsymbol{t}), \boldsymbol{s}^*$ are the current feature vector and the desired feature vector with zero derivatives, respectively. Changes in s depend on only camera motion. Furthermore, IBVS assumes that the camera is attached to a six-DOF system such as a robot manipulator with eye-in-hand configuration and $\boldsymbol{k} \geq \mathbf{6}$.

Instead of point features, image moments were preferred as the shape features for this study to more precisely define the target of the quadrotor. The contours of an image's form can be characterized using image moments $(\boldsymbol{m_{ij}})$, which are defined in two dimensions as

$$\boldsymbol{m_{ij}} = \sum_{\boldsymbol{k_c}=\mathbf{1}}^{\boldsymbol{n}} \boldsymbol{x_c^i} \cdot \boldsymbol{y_c^j} \tag{2}$$

where $\boldsymbol{k_c}$ represents the image contour points' index, $(\boldsymbol{x_c}, \boldsymbol{y_c})$ are the contour points' coordinates, $\boldsymbol{n}$ is the number of contour pixels, and $\boldsymbol{i}, \boldsymbol{j}$ are the number of contour pixels' powers. To ensure robustness against shape-scaling, it is necessary to normalize these moments. The centered moments $\boldsymbol{\mu_{ij}}$, which are normalized moments, are defined as follows:

$$\boldsymbol{\mu_{ij}} = \sum_{\boldsymbol{k_c}=\mathbf{1}}^{\boldsymbol{n}} \left(\boldsymbol{x_c} - \boldsymbol{x_g}\right)^{\boldsymbol{i}} \cdot \left(\boldsymbol{y_c} - \boldsymbol{y_g}\right)^{\boldsymbol{j}} \tag{3}$$

where the coordinates of the image centroid are defined as $\boldsymbol{x_g} = \boldsymbol{m_{10}}/\boldsymbol{n}$ and $\boldsymbol{y_g} = \boldsymbol{m_{01}}/\boldsymbol{n}$. These moments in (2) and (3) can be used to create moments that provide scale, rotation, and translation invariance.

In quadrotors, 6 DOFs are controlled by 4 thrusters, resulting in underactuation of the vehicle. The linear motion in $(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z})$ and rotation around the $\boldsymbol{z}$-axis, which results in 4 DOFs, were taken into consideration to relate each feature to a DOF. Please note that this is only acceptable under small roll and pitch maneuvers. When a planar target and the image plane are parallel, as illustrated in Figure 1, the definition of perspective projection moments with scaling for controlling 4 DOFs is as follows:

$$\boldsymbol{a_n} = \boldsymbol{Z^*}\sqrt{\frac{\boldsymbol{a^*}}{\boldsymbol{a}}}, \boldsymbol{x_n} = \boldsymbol{a_n x_g}, \boldsymbol{y_n} = \boldsymbol{a_n y_g}, \boldsymbol{\theta_n} = \left(\frac{\mathbf{1}}{\mathbf{2}}\boldsymbol{atan}\left(\frac{\mathbf{2}\boldsymbol{\mu_{11}}}{\boldsymbol{\mu_{20}} - \boldsymbol{\mu_{02}}}\right)\right)/\boldsymbol{\pi} \tag{4}$$

$$\boldsymbol{s} = \begin{bmatrix} \boldsymbol{x_n} & \boldsymbol{y_n} & \boldsymbol{a_n} & \boldsymbol{\theta_n} \end{bmatrix}^{\boldsymbol{T}} \tag{5}$$

where $\boldsymbol{a_n}$ is the normalized area of the closed contour shape, $\boldsymbol{a^*}$ is the desired target area and $\boldsymbol{Z^*}$ is the desired target depth with $\boldsymbol{s}$ as the feature vector. $(\boldsymbol{x_n}, \boldsymbol{y_n})$ are the normalized image center coordinates and $\boldsymbol{\theta_n}$ is the normalized shape orientation. The proposed IBVS system's feature vector is gathered from these features.

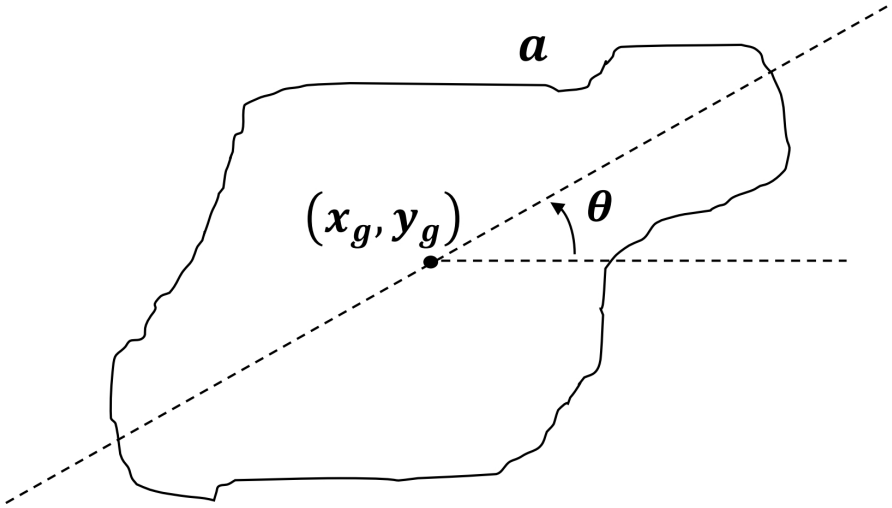


Figure 1. A closed contour shape with perspective projection moments.

The interaction matrix $\boldsymbol{L_s}$ is a matrix that represents the connection between the feature error vector $\boldsymbol{e}$ and the image dynamics. The above-mentioned visual characteristics' interaction matrix is provided as in [3]:

$$\dot{\boldsymbol{s}} = L_s \cdot \boldsymbol{v} \Rightarrow \dot{\boldsymbol{s}} = \begin{bmatrix} -1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & -1 \end{bmatrix} \begin{bmatrix} \nu_x \\ \nu_y \\ \nu_z \\ \omega_z \end{bmatrix} \tag{6}$$

As shown in Equation (6), $\boldsymbol{L_s}$ is equal to $-\boldsymbol{I_{3\times 3}}$ for these moment features [5]. As a result of the chosen 4 DOFs, the terms of (ω_x, ω_y) in $\boldsymbol{v}$ velocity vector were omitted in Equation (6). Furthermore, as a physical limitation of quadrotors that are underactuated systems, this definition is only valid when the quadrotor avoids aggressive maneuvers [10]. After the interaction matrix representation, the relation between the error and the vehicle velocities was obtained using Equations (6) and (1) with fixed desired features, as follows:

$$\dot{\boldsymbol{e}} = L_s \cdot \boldsymbol{v} \tag{7}$$

IBVS utilizes a differential equation to exponentially reduce the feature error.

$$\dot{\boldsymbol{e}} + \lambda \cdot \boldsymbol{e} = \boldsymbol{0} \tag{8}$$

From a different perspective, this differential equation serves as the primary equation to assign a sliding surface in conventional SMC. Considering the errors, their derivatives and the system degree, conventional SMC specifies a surface, and it attempts to maintain the states of the system by equalizing the surface to zero. Under this concept, SMC and IBVS exhibit a close relationship. IBVS defines a kinematic velocity controller and is defined for quadrotors with the Moore–Penrose pseudoinverse of the estimated interaction matrix $\boldsymbol{L_s^+}$, which is the same as itself and the gain $\boldsymbol{\lambda}$ using Equations (1) and (6)–(8) as:

$$\boldsymbol{v} = -\boldsymbol{L_s^+} \cdot \boldsymbol{\lambda} \cdot \boldsymbol{e} = \boldsymbol{\lambda} \cdot \boldsymbol{e} \tag{9}$$

The IBVS controller for quadrotors in Equation (9) has drawbacks, such as a long convergence time with fixed gain or sudden change in the initial velocity values. The sliding surface designs were deployed in the proposed system to avoid these drawbacks. The proposed IBVS system with sliding surface designs is shown in Figure 2 and the sliding surface designs are defined in detail in the following subsections.

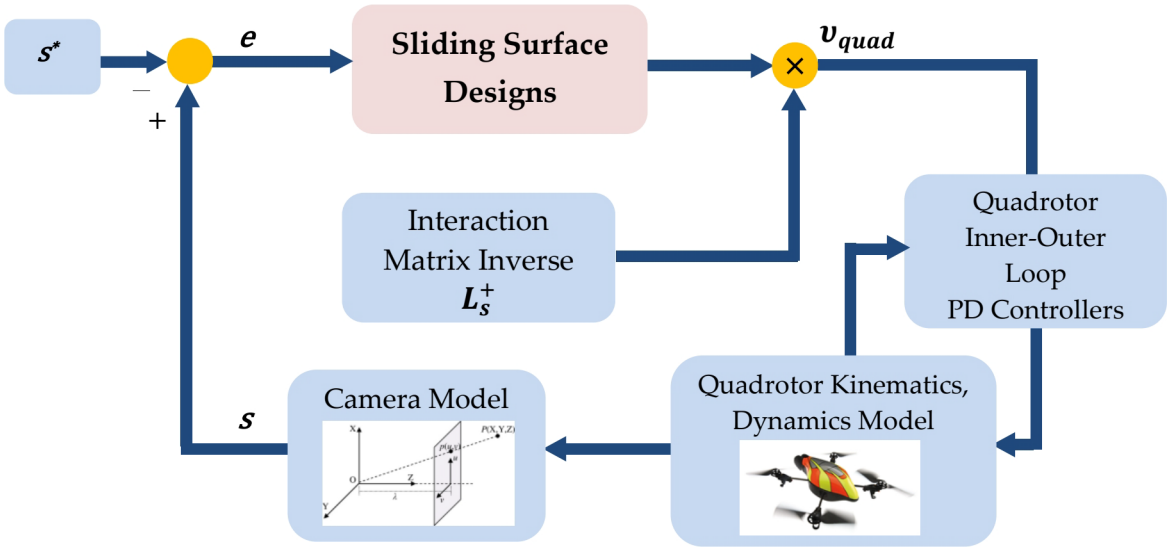


Figure 2. The schematic of the proposed IBVS system with sliding surface designs and system dynamics.

2.1. Fuzzy Logic for Linear Varying Sliding Surfaces

The IBVS system proposed in this study aimed to achieve three VS control performance measures. The first one is the feature convergence time, which is a measure of the time interval for the stabilization or autolanding of a quadrotor according to the visual feature errors [28]. The quadrotor's velocity restrictions should be taken into account in addition to aiming for success in this measure. A commonly used technique in SMC involves modifying the sliding slope in response to the system error states to achieve the desired objective while ensuring that the control signal remains within specified limits. The sliding slope is the only design parameter in Equations (8) and (9), and an online tuning algorithm was proposed, using the varying sliding slope approach. Instead of utilizing a complex analytical technique that has to be fine-tuned to account for soft parameter transition and makes use of numerous parameters to establish a parameter, FL can be a suitable option to add linguistic meanings and user experience to the design while also avoiding the requirement for repetitive tuning. The IBVS velocity controller in Equation (9) was modified as in Equation (10):

$$v_{quad} = \lambda_{FL}(\|e\|, d\|e\|/dt) \cdot e \tag{10}$$

λ_{FL} should be dependent on $(\|e\|, d\|e\|/dt)$, which determines VS velocity profiles. The rule base of the FL unit was specified in accordance with gain scheduling behaviors, which result in low velocities when there are high values of $(\|e\|, d\|e\|/dt)$ and vice versa. The rule members of this rule base are defined by considering the trials and user experience that have been described in [29]. The evaluation of surfaces relies on multiple factors to determine the relationships between inputs and outputs. These are the types of membership functions (MF) utilized for the signals, the method of aggregating MFs, the rule base, and the type of defuzzification. Mamdani, as the type of FL unit used in this study, utilized fuzzy MFs as the output functions. The input–output MFs were Gaussian and generalized bell type. The aggregation type was the maximum of MFs, and the centroid of area (COA), the type of defuzzification, was the weighted average of the centroids of the output MFs with weighting factors of the input MFs and rules.

2.2. Fixed Accelerating Sliding Surface with Time Variation

In Ref. [30], Bartoszewicz proposed a sliding mode approach with time optimality for the robust control of second-order uncertain systems. This optimality provides an adaptation of the system to the zero initial conditions that are essential for an IBVS system. A sliding slope term with time variation that was added to the classical sliding surface was defined

as a function of time and was active during a predefined time interval. Bartoszewicz also proposed two sliding surfaces that have fixed acceleration and fixed velocity, which are defined by their time dependency. Compared to the conventional sliding surface, the sliding surface with fixed acceleration under time variations offers a higher error convergence; hence, this design was selected. Fixed accelerating sliding surfaces with time variation, parameter definitions, and an IBVS control law are given in Equations (11)–(13), respectively:

$$\dot{e} + \lambda \cdot e + \begin{cases} A \cdot t^2 + B \cdot t + C & for\ t \leq T \\ 0 & for\ t > T \end{cases} = 0 \tag{11}$$

$$\begin{aligned} A &= -\frac{\dot{e}(0) + \beta \cdot e(0)}{T^2} \\ B &= 2\frac{\dot{e}(0) + \beta \cdot e(0)}{T} \\ C &= -\dot{e}(0) - \beta \cdot e(0) \end{aligned} \tag{12}$$

$$v_{quad} = \lambda \cdot e + \begin{cases} A \cdot t^2 + B \cdot t + C & for\ t \leq T \\ 0 & for\ t > T \end{cases} = 0 \tag{13}$$

where β is the initial error constant, T is the time limit for fixed acceleration, and A, B and C are second-order time polynomial coefficients.

2.3. Nonlinear Sliding Surface with Tangent Hyperbolic Function

The first two designs' sliding surfaces are linear; therefore, the error states will always try to reach a constant sliding slope. The sliding surfaces with linear variations or time variations may reveal piecewise linear surfaces. To shorten the reach and slide phases, it is possible to employ a nonlinear function as the sliding surface function, as an alternative to the conventional linear approaches [31]. Tangent hyperbolic functions are strong nonlinear mapping functions and were chosen as the candidate in this study. In Equation (14), the error dynamics are defined using this nonlinear function as follows:

$$\dot{e} + w_p(e) \cdot \mathbf{tanh}(c_i \cdot e) = 0 \tag{14}$$

where w_p is the parameter for sliding magnitude that denotes the convergence rate and c_i represents the nonlinear surface's sliding slope. These parameters might be set to fixed values; however, to reach the sliding surface more quickly, w_p is selected as a function of error. The velocity signals of VS with this design are defined as:

$$v_{quad} = w_p(e) \cdot \mathbf{tanh}(c_i \cdot e) \tag{15}$$

2.4. Fuzzy Logic for Integral Sliding Surfaces

The classical linear sliding surface design in Equation (8) is a PD-type sliding surface and an integral term was added to this design to increase tracking performance [32]. This term may be in effect for all error trajectories, or only when the errors fall within predetermined boundaries based on the limits of the control signal. This is to avoid the windup effect of the integral term. The integral sliding surface definition is given below:

$$\lambda_p \cdot e + \lambda_i \cdot \int e \cdot dt + \dot{e} = 0 \tag{16}$$

The main problem in this surface design is determining the appropriate gain values for (λ_p, λ_i). According to the information provided in [33], FL is also suitable for this design. In this study, only the gain value for the integral term was assigned using FL as a function of $\|e\|$. The FL characteristics were the same as those mentioned in the design of linear varying sliding surfaces with fuzzy logic. Equation (16) was reformulated for IBVS as:

$$v_{quad} = \lambda_p \cdot e + \lambda_{i-FL}(\|e\|) \cdot \int e \cdot dt \tag{17}$$

2.5. Nonlinear Sliding Surface with Time Variation

As an alternative to nonlinear constant functions, as shown in Equation (14), nonlinear time-varying surfaces can be designed to shorten the reaching phase of the states. Exponential time-varying functions were proposed in the literature as the nonlinear time-varying sliding surface function candidates, and the design in [34] was chosen in this study as:

$$\dot{\boldsymbol{e}} + \boldsymbol{\lambda_1}\cdot\boldsymbol{e} - \left[\dot{\boldsymbol{e}}(0) + \boldsymbol{\lambda_2}\cdot\boldsymbol{e}(0)\right]\cdot e^{-kt} = \mathbf{0} \tag{18}$$

where $(\boldsymbol{\lambda_1}, \boldsymbol{\lambda_2}, \boldsymbol{k})$ are the design parameters of the sliding slope and time constant of convergence, respectively. This function also included the initial values of error and error derivatives. This provided zero initial velocity signals for IBVS. The control law for IBVS is given as follows:

$$\boldsymbol{v_{quad}} = \boldsymbol{\lambda_1}\cdot\boldsymbol{e} - \left[\dot{\boldsymbol{e}}(\mathbf{0}) + \boldsymbol{\lambda_2}\cdot\boldsymbol{e}(\mathbf{0})\right]\cdot\boldsymbol{e}^{-kt} \tag{19}$$

After these design phases of the proposed system, the velocities were then applied to the quadrotor as indicated in Figure 2. The quadrotor's inner–outer loop PD controllers regulated each rotor's torque, and the quadrotor moved in accordance with its kinematics and dynamics. A new image was captured by the quadrotor's camera, completing the closed loop of the system presented in Figure 2.

3. Simulation Results

This study utilized MATLAB Simulink, Robotics Toolbox, Machine Vision Toolbox [35], Fuzzy Logic Toolbox, and CoppeliaSim (version 4.5.1) to simulate and animate the proposed IBVS system with the sliding surface designs. The quadrotor model utilized here is the X-4 flyer, and more information about it can be found in [36]. It is considered that a camera with a downward FOV is fixed to the quadrotor's center without any kinematic transformation. The camera has a 1024 × 1024 pixel resolution, and the principal point coordinates are defined as (512,512) in the image plane. The system's control loop and video stream both operate at 20 Hz. The presumed center of the target shape, denoted as "H", is located at [−0.025 0.375 0] m. The closed contour of the target shape is obtained using Moore's neighborhood tracking method and that contour information was used to extract shape features. Figure 3 shows the animated scene of the scenarios created in CoppeliaSim.

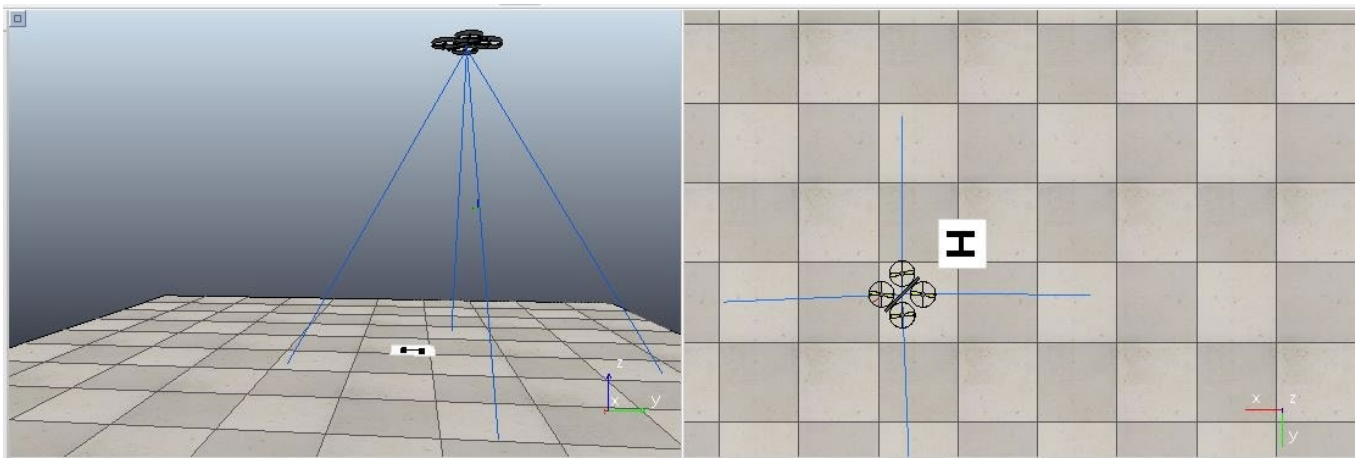


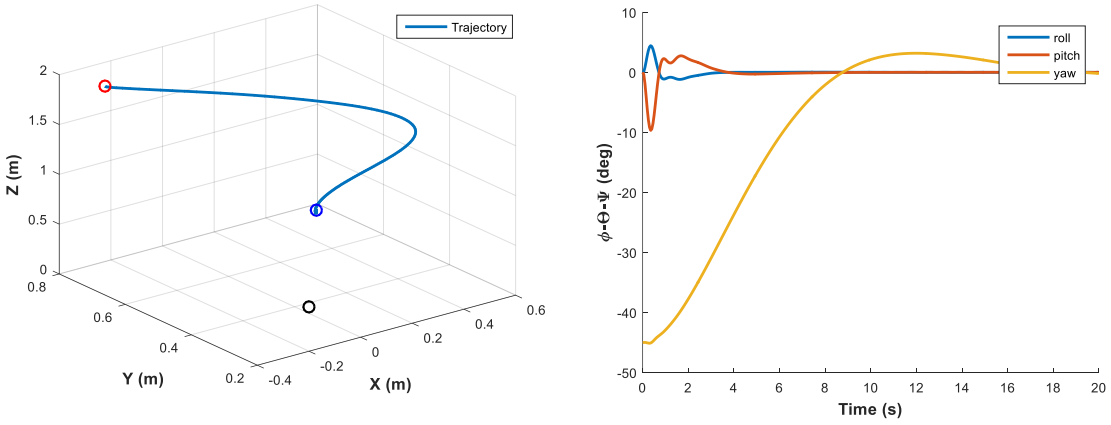
Figure 3. The animated scene of VS scenario in CoppeliaSim.

Details and parameters for three IBVS sliding surface designs, including the conventional IBVS, are provided in the following subsections, along with examples of the results. To demonstrate the designs' robustness, the shape's features are perturbed by uniformly distributed random noises with a standard deviation of 3 for $\left(\boldsymbol{x_g}, \boldsymbol{y_g}, \boldsymbol{\theta}\right)$ and 10 for $\boldsymbol{a}$, which may occur due to factors such as camera disturbance or wind affecting the quadrotor [37].

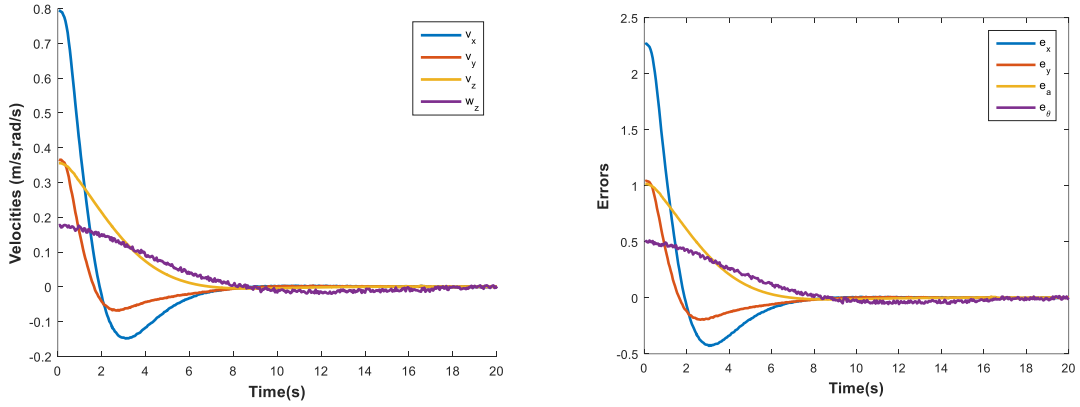
The initial positions and orientations of the quadrotor are $(X_0, Y_0, Z_0) = \begin{bmatrix} -0.35 & 0.7 & 2 \end{bmatrix}^T$ m. and $(\phi_0, \theta_0, \psi_0) = \begin{bmatrix} 0 & 0 & -\pi/4 \end{bmatrix}^T$ rad., respectively. The target features are $x_n^* = y_n^* = 512, Z^* = 1.2$ m., $\theta_n^* = 0^o$ with $(X_T, Y_T, Z_T) = \begin{bmatrix} -0.025 & 0.375 & 0 \end{bmatrix}^T$ m. The first performance metric is the convergence time, which is the time instant when $\|e\| < 0.03$. The second parameter, named the arc length, represents the length of the path followed by the quadrotor. This can be a good sign of the consumed energy and is calculated using the curve length formulation in vector calculus [38].

3.1. Classical IBVS with Fixed Sliding Slope

To compare each design, a conventional IBVS system using a fixed slope is considered as in Equation (9) and the slope is chosen as 0.35. Figure 4 illustrates the results of this conventional system. Figure 4a shows the quadrotor's trajectory, with the red circle representing the initial point, the blue circle representing the completion point, and the black circle representing the target center. Figure 4b shows the RPY angles of the quadrotor in the defined scenario. It is clear that there are no aggressive maneuvers and the limitation in Equation (6) is provided. Moreover, the velocity and error profiles are quite similar, as expected in Equation (6). The velocities and the errors in Figure 4c,d contain the characteristics of the disturbance. The convergence time is 16.25 s and the arc length is 1.7336 m for this approach.



(**a**) 3D trajectory of the quadrotor (**b**) RPY angles



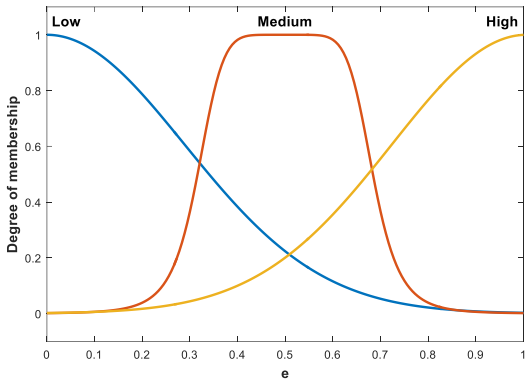
(**c**) Velocities (**d**) Feature errors

Figure 4. Results for classical IBVS.

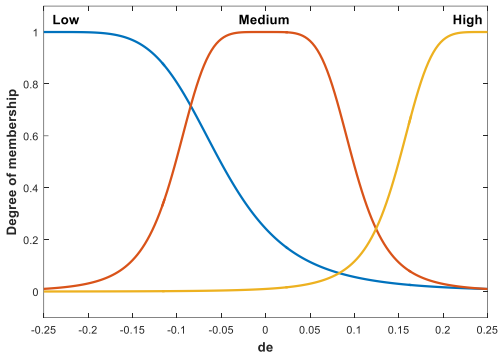
3.2. Design 1: IBVS with Fuzzy Logic for Linear Varying Sliding Surfaces

To achieve fast convergence and low velocity profiles, a linear varying sliding slope design was presented as the first design. FL, also known as the key for the fuzzy sliding mode, is proposed to assign the slope value according to the error dynamics. The normal-

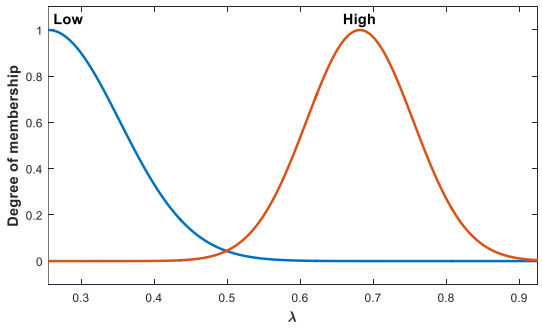
ized error and error derivative inputs are applied to the FL unit. Mamdani is the type of FL unit and the input–output MFs are Gaussian and generalized bell type, as shown in Figure 5a–c. The aggregation type is the maximum of MFs and type of defuzzification is COA. The rulebase of the FL unit is also given in Table 1. The rulebase aims to provide a low gain while $\|e\|$ and $d\|e\|/dt$ are high to avoid high velocities. Subsequently, the gain is increased to provide fast convergence while $\|e\|$ and $d\|e\|/dt$ are low. Fine-tunings of membership functions and the rulebase are implemented by trial and error according to the nature of FL. The surface of the FL unit according to defined membership functions for input–output and the rulebase is shown in Figure 5d. This nonlinear surface is a result of nonlinear Gaussian and generalized bell input–output MFs. Here, it should be noted that λ_{FL} is limited between 0.33 and 0.68.



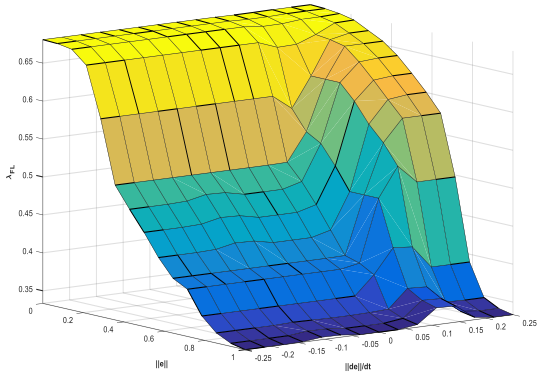
(**a**) Membership functions for $\|e\|$



(**b**) Membership functions for $d\|e\|/dt$



(**c**) Membership functions for λ_{FL}



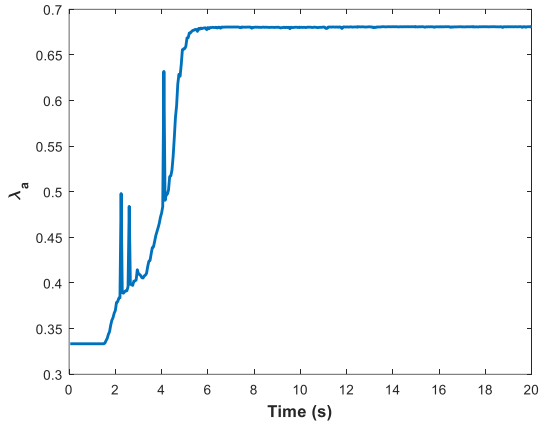
(**d**) FL surface for λ_{FL}

Figure 5. FL membership functions and surface for Design 1.

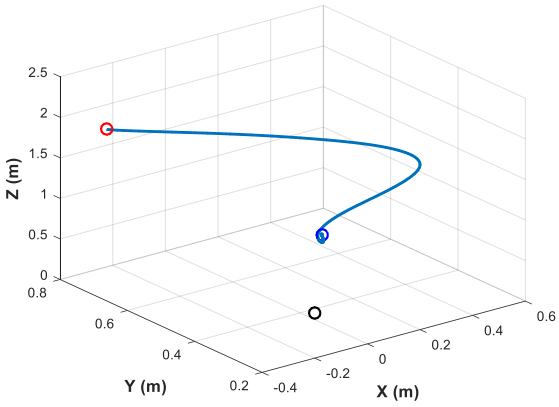
Table 1. FL rulebase for λ_{FL}.

$\|e\|$, $d\|e\|/dt$	**Low**	**Medium**	**High**
Low	High	High	High
Medium	Low	Low	High
High	Low	Low	Low

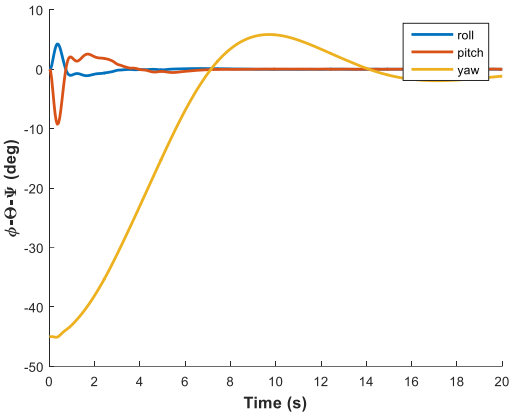
λ_{FL} remains constant and low between 0 and 1.5 s, since the error and error derivative norms are high, as in Figure 6a. Then, the FL unit is activated by error and error derivative norms and the varying sliding slope shows a nonlinear characteristic without discontinuity. Again, it remains constant and high after norms are small with small fluctuations after 6.2 s. These fluctuations are consequences of noise affecting the features. They are also obvious in the velocity signals of the quadrotor in Figure 6d, which may result in aggressive maneuvers.



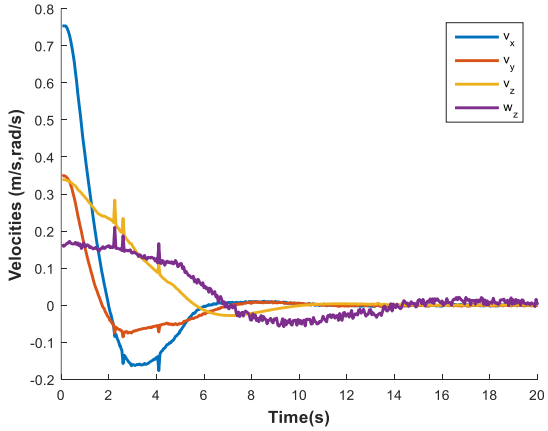
(**a**) FL output for sliding surface



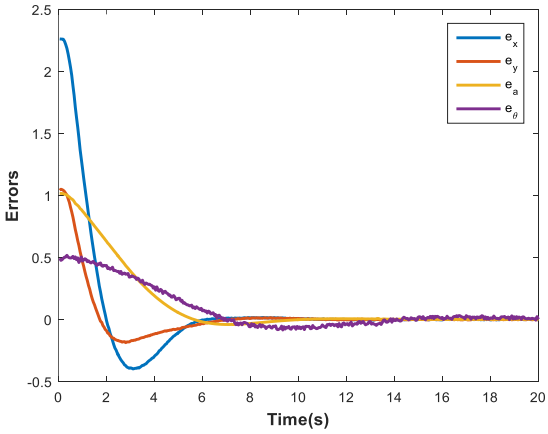
(**b**) 3D trajectory of the quadrotor



(**c**) RPY angles



(**d**) Velocities



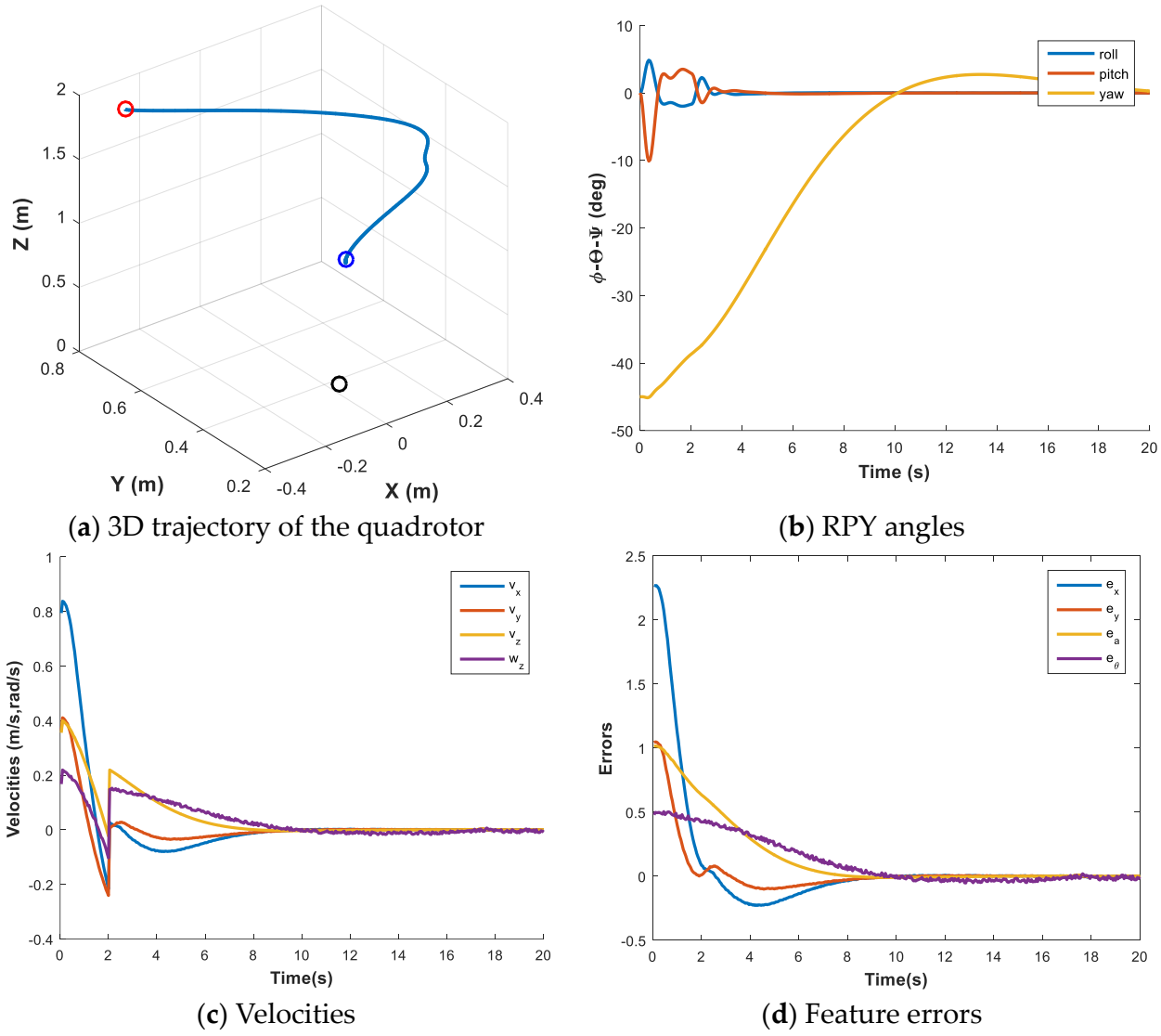
(**e**) Feature errors

Figure 6. Results for Design 1.

The convergence time is 12.95 s. and the arc length is 1.8009 m. The design demonstrates faster convergence, as expected. However, it is worth noting that the arc length is slightly longer due to the varying sliding slope.

3.3. Design 2: IBVS with Fixed Accelerating Sliding Surface with Time Variations

This design added a new sliding slope term as a function of time, which can be active for a predefined time interval in Equations (11) and (12). Constant accelerated sliding was also chosen in this design. β and T are defined as 0.22 and 2 by trials, respectively. The results of this design are shown in Figure 7.



(**a**) 3D trajectory of the quadrotor (**b**) RPY angles

(**c**) Velocities (**d**) Feature errors

Figure 7. Results for Design 2.

In Figure 7a,c, the effect of the constant acceleration term is quite clear. The impact of the initial conditions of the velocity signals is significant, particularly when they deviate considerably from zero. This method aims to reduce these effects specifically within the time range of 0–2. The quadrotor follows a different path and shows different velocity patterns after this term is disabled. The sudden change in velocities at 2 s is very dramatic; however, again, it is compensated by inner loop controllers. Here, it should be noted that e_y increases of 2–3.5 s are caused by the dragging force of the time optimal sliding surface but IBVS gathers itself up.

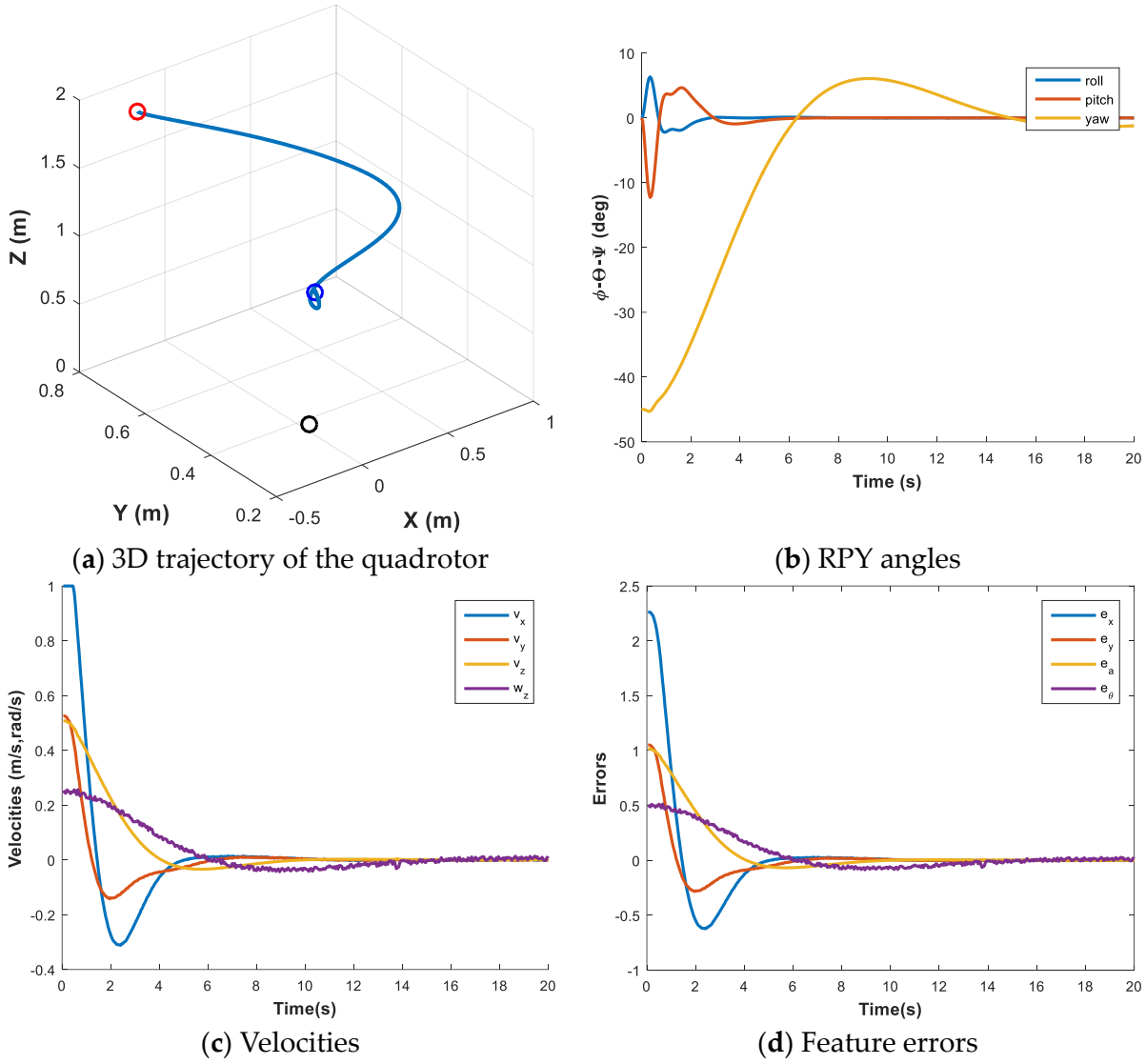
This design also provides a faster convergence with a 9.05 s. convergence time. The arc length is 1.6269 m, which is the shortest one among the designs.

3.4. Design 3: IBVS with Nonlinear Sliding Surface with Tangent Hyperbolic Function

Nonlinear sliding surfaces may shorten the reach phase and convergence time. The tangent hyperbolic function in Equation (14) with error norm dependency is defined as

$$\begin{aligned} w_p(e) &= -\mathbf{0.02}\|e\| - \mathbf{50} \\ c_i &= \mathbf{0.01} \end{aligned} \tag{20}$$

Here, it should be noted that w_p can be chosen as a function of time [31]. The results of this design are shown in Figure 8.



(**a**) 3D trajectory of the quadrotor (**b**) RPY angles

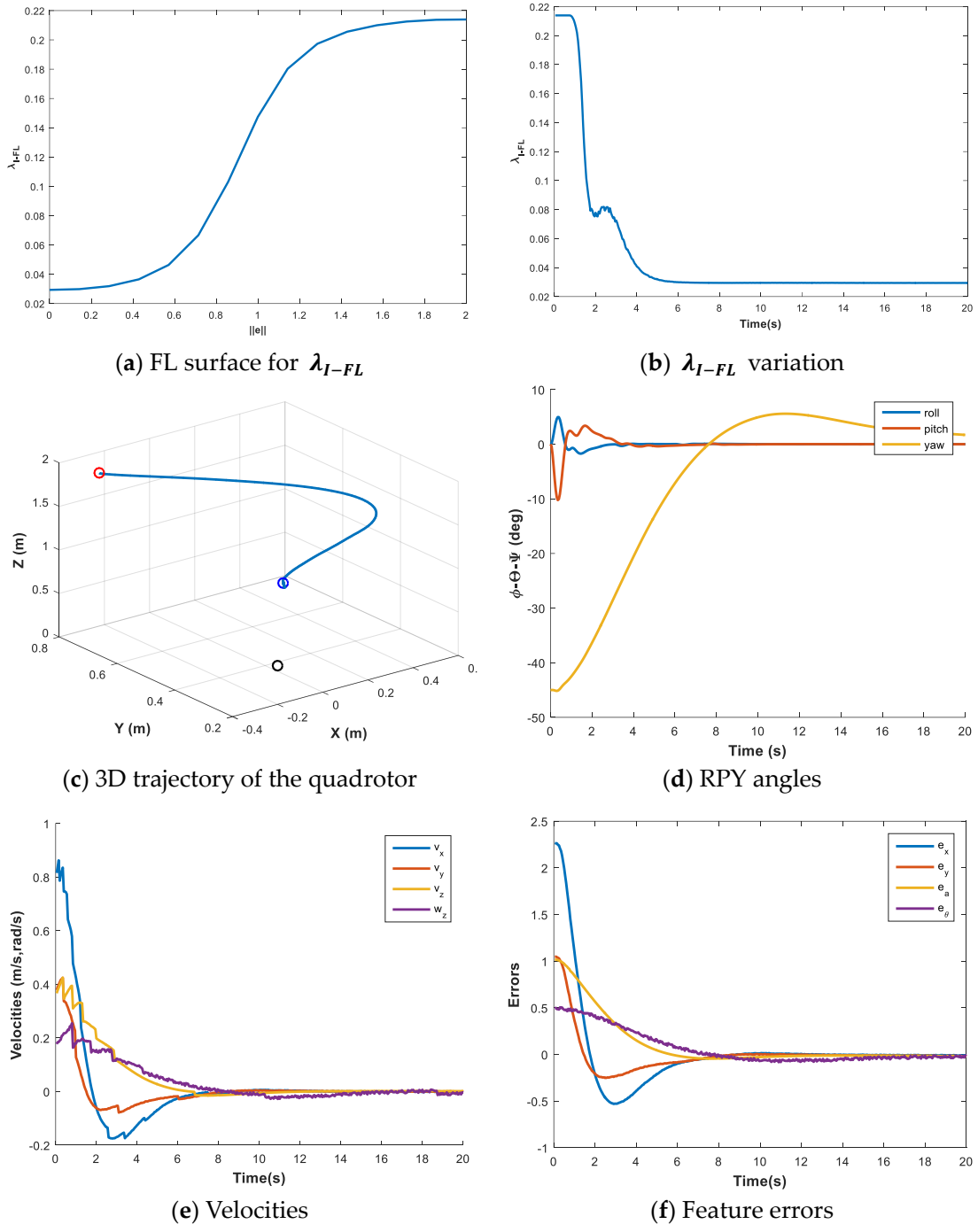
(**c**) Velocities (**d**) Feature errors

Figure 8. Results for Design 3.

The results in Figure 8 show that this design forces the IBVS system to converge more rapidly; however, the velocity limits do not allow for this. In Figure 8c, the v_x signal in blue starts above the velocity limit of 1 m/s. Moreover, it is observed that the magnitudes of the negative peaks in velocities and errors are larger in comparison to the other approaches. This suggests that the nonlinear surface causes the features to drag from the desired target. In contrast, the observed convergence time is approximately 12.65 s, a value that closely aligns with the expected result of the linear varying surface design. The arc length is 1.97 m, which is a consequence of larger velocity and error magnitudes.

3.5. Design 4: IBVS with Fuzzy Logic for Integral Sliding Surfaces

The integral sliding mode with FL in Equation (17) was proposed to increase tracking performance and the results are shown in Figure 9. λ_p is assigned as 0.35 and λ_{I-FL} uses $\|e\|$ as input. Again, the type of the FL unit of λ_{I-FL} is Mamdani and two Gaussian membership functions are defined for both input and output with the surface shown in Figure 9a.



(**a**) FL surface for $\boldsymbol{\lambda_{I-FL}}$ (**b**) $\boldsymbol{\lambda_{I-FL}}$ variation

(**c**) 3D trajectory of the quadrotor (**d**) RPY angles

(**e**) Velocities (**f**) Feature errors

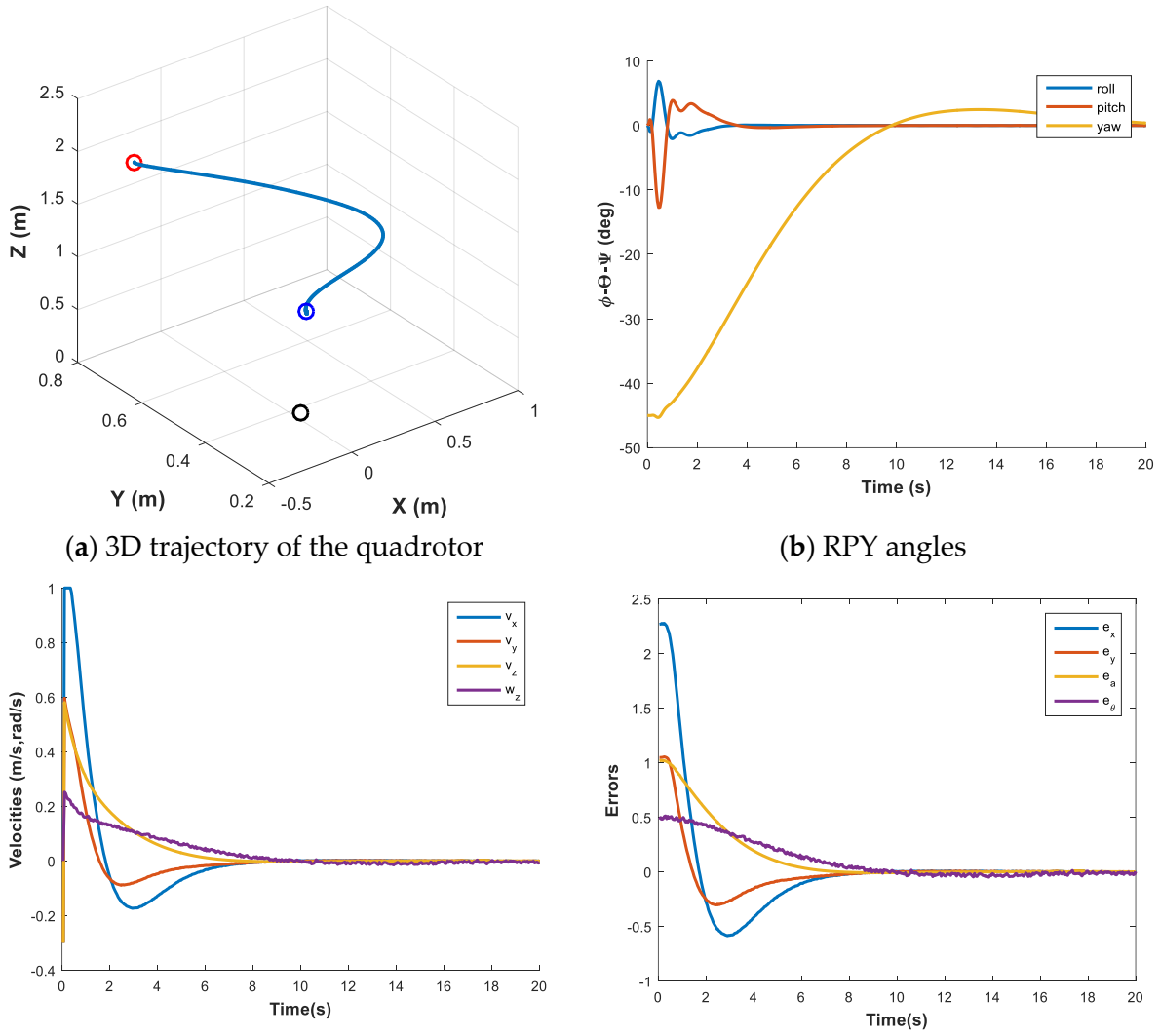
Figure 9. Results for Design 4.

λ_{I-FL} variations for the same scenario are shown in Figure 9b and it is clear that it approximates to nearly zero to avoid the integral windup while the errors and the error norm are observed to be very close to zero, as shown in Figure 9f. The velocity signals show abrupt changes in the first 4 s in Figure 9e. Fortunately, they are filtered effectively by the inner loop dynamic controllers of the quadrotor.

The convergence time for this design is 17.3 s. This is worse than the classical approach. It is clear that the integral term forces fast convergence with high velocities. Please note that relying only on the integral term is inadequate to achieve a fast convergence in the VS system. The arc length is also 1.83 m., indicating a reduction in comparison to the previous design.

3.6. Design 5: IBVS with Nonlinear Sliding Surface with Time Variation

An alternate approach to nonlinear sliding surfaces involves changing the nonlinear function as a function of time. An exponential function defined in Equation (18) is defined with λ_1, λ_2 and k values as 0.3, 0.2 and 2, respectively. The results of this design are shown in Figure 10.



(**a**) 3D trajectory of the quadrotor (**b**) RPY angles

(**c**) Velocities (**d**) Feature errors

Figure 10. Results for Design 5.

The path in Figure 10a is very smooth and the RPY signals are quite realizable. Again, v_x signal in blue starts with saturation for a short time interval. In Figure 10c, it is also clear that the nonlinear time-varying design provides zero initial velocities, which are essential for realization. Furthermore, the system behaves like the classical system after the nonlinear term is disabled after 2 s in Equation (18).

The convergence time is 8.65 s, which is the shortest among the designs. However, the arc length is 1.8237 m, which can be assumed to be the average.

4. Conclusions

While new studies try to adapt kinematic controller approaches of vs. to UAV applications, sliding surface designs may provide expert help in the field of nonlinear control. In this study, five different sliding surface designs with analytical and intelligent methods are modified and an IBVS system with these designs for quadrotors is proposed. The designs

are compared according to the convergence time, arc length, and motion characteristics under feature disturbances. The issues in realization, such as initial velocity values, are also discussed. A comparison of the designs is given in Table 2. According to the convergence time metric, it is evident that all designs perform better than the classical approach except the integral sliding surface with fuzzy logic. The best one is the fifth design, the nonlinear sliding surface with time variations, which is 46.77% shorter than the classical approach. The best one according to the path length, which can be a good marker of the energy consumed by the quadrotor, is the second design, the time-varying sliding surface with a fixed acceleration that is 6.15% shorter. In contrast, this solution activates the constant acceleration term within a specific duration, resulting in sudden changes in the velocity signals. However, the inner loop controller effectively mitigates this undesired effect. According to motion characteristics, Designs 1 and 4 also have the potential for aggressive maneuvers. It is important to note that parameter tuning is a necessary aspect of all designs, which is a function of experience. As a limitation of Equation (6), the quadrotor must show soft maneuvers and these are provided in all designs with small roll and pitch angles.

Table 2. A comparison of the designs.

	Parameters	Feature Conv. Time (s)	Path Length (m)	Maneuver Type	Computational Cost
Classical	λ	16.25	1.7336	Soft	Low
Design 1	λ_{FL}	12.95	1.8009	Aggressive (possible)	Medium
Design 2	λ, A, B, C, T	9.05	1.6269	Aggressive (possible)	Low
Design 3	$w_p(\cdot), c_i$	12.65	1.97	Soft	Low
Design 4	$\lambda_P, \lambda_{I-FL}$	17.3	1.83	Aggressive (possible)	Medium
Design 5	λ_1, λ_2, k	8.65	1.8237	Soft	Low

The computational costs of all designs are advantageous for embedded on-board UAV systems considering the perspective of realization. The use of FL units may potentially result in a bottleneck when deploying a UAV with limited computational capacity, such as the STM32. Moreover, it is worth considering the use of AI structures such as ANFIS, which combines NN and FL, as an alternative to FL units. This approach may have the potential to enhance the performance of designs in a more effective manner. In future studies, it is planned to realize the proposed sliding surface designs on a real quadrotor that is controlled by visual servoing. Furthermore, the study utilizes point features. Using shape features, the features of an AR code, a popular marker in the field of VS, may be more attractive. Besides these sliding mode designs, modifications to nonlinear time variances and terminal sliding mode will be addressed and tested to enhance the number of designs. Additionally, all the designs will be tested after adding a type of Kalman filter while tracking moving feature targets.

Funding: This research received no external funding.

Data Availability Statement: Not applicable.

Conflicts of Interest: The author declares no conflict of interest.

References

1. Roy, B.; DasGupta, A.; Paul, A. Impact of Space Weather Events on Satellite-Based Navigation. *Sp. Weather* **2013**, *11*, 680–686. [CrossRef]
2. Chaumette, F.; Hutchinson, S. Visual Servo Control. I. Basic Approaches. *IEEE Robot. Autom. Mag.* **2006**, *13*, 82–90. [CrossRef]
3. Tahri, O.; Chaumette, F. Point-Based and Region-Based Image Moments for Visual Servoing of Planar Objects. *IEEE Trans. Robot.* **2005**, *21*, 1116–1127. [CrossRef]
4. Bourquardez, O.; Mahony, R.; Hamel, T.; Chaumette, F. Stability and Performance of Image Based Visual Servo Control Using First Order Spherical Image Moments. In Proceedings of the 2006 IEEE/RSJ International Conference on Intelligent Robots and Systems, Beijing, China, 9–15 October 2006; pp. 4304–4309. [CrossRef]

5. Chaumette, F.; Hutchinson, S. Visual Servo Control. II. Advanced Approaches. *IEEE Robot. Autom. Mag.* **2007**, *14*, 109–118. [CrossRef]
6. Yuksel, T. Intelligent Visual Servoing with Extreme Learning Machine and Fuzzy Logic. *Expert Syst. Appl.* **2017**, *72*, 344–356. [CrossRef]
7. Chesi, G.; Hashimoto, K.; Prattichizzo, D.; Vicino, A. Keeping Features in the Field of View in Eye-in-Hand Visual Servoing: A Switching Approach. *IEEE Trans. Robot.* **2004**, *20*, 908–913. [CrossRef]
8. Hamel, T.; Mahony, R. Visual Servoing of an Under-Actuated Dynamic Rigid-Body System: An Image-Based Approach. *IEEE Trans. Robot. Autom.* **2002**, *18*, 187–198. [CrossRef]
9. Mahony, R.; Hamel, T. Image-Based Visual Servo Control of Aerial Robotic Systems Using Linear Image Features. *IEEE Trans. Robot.* **2005**, *21*, 227–239. [CrossRef]
10. Bourquardez, O.; Mahony, R.; Guenard, N.; Chaumette, F.; Hamel, T.; Eck, L. Kinematic Visual Servo Control of a Quadrotor Aerial Vehicle. *IEEE Trans. Robot.* **2007**, *25*, 833–838.
11. Hamel, T.; Mahony, R. Image Based Visual Servo-Control for a Class of Aerial Robotic Systems. *Automatica* **2007**, *43*, 1975–1983. [CrossRef]
12. Altug, E.; Ostrowski, J.P.; Taylor, C.J. Control of a Quadrotor Helicopter Using Dual Camera Visual Feedback. *Int. J. Rob. Res.* **2005**, *24*, 329–341. [CrossRef]
13. Ceren, Z.; Altug, E. Image Based and Hybrid Visual Servo Control of an Unmanned Aerial Vehicle. *J. Intell. Robot. Syst.* **2012**, *65*, 325–344. [CrossRef]
14. Metni, N.; Hamel, T. Visual Tracking Control of Aerial Robotic Systems with Adaptive Depth Estimation. *Int. J. Control Autom. Syst.* **2007**, *5*, 10.
15. Sun, L.; Huang, Y.; Zheng, Z.; Zhu, B.; Jiang, J. Adaptive Nonlinear Relative Motion Control of Quadrotors in Autonomous Shipboard Landings. *J. Frankl. Inst.* **2020**, *357*, 13569–13592. [CrossRef]
16. 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]
17. Cunha, R.; Silvestre, C.; Hespanha, J.; Pedro Aguiar, A. Vision-Based Control for Rigid Body Stabilization. *Automatica* **2011**, *47*, 1020–1027. [CrossRef]
18. Herissé, B.; Hamel, T.; Mahony, R.; Russotto, F.-X. Landing a VTOL Unmanned Aerial Vehicle on a Moving Platform Using Optical Flow. *IEEE Trans. Robot.* **2012**, *28*, 77–89. [CrossRef]
19. Plinval, H.; Morin, P.; Mouyon, P.; Hamel, T. Visual Servoing for Underactuated VTOL UAVs: A Linear, Homography-Based Framework. *Int. J. Robust Nonlinear Control* **2014**, *24*, 2285–2308. [CrossRef]
20. 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]
21. Thomas, J.; Loianno, G.; Daniilidis, K.; Kumar, V. Visual Servoing of Quadrotors for Perching by Hanging From Cylindrical Objects. *IEEE Robot. Autom. Lett.* **2016**, *1*, 57–64. [CrossRef]
22. Mebarki, R.; Lippiello, V.; Siciliano, B. Nonlinear Visual Control of Unmanned Aerial Vehicles in GPS-Denied Environments. *IEEE Trans. Robot.* **2015**, *31*, 1004–1017. [CrossRef]
23. Shi, H.; Hwang, K.S.; Li, X.; Chen, J. A Learning Approach to Image-Based Visual Servoing with a Bagging Method of Velocity Calculations. *Inf. Sci.* **2019**, *481*, 244–257. [CrossRef]
24. Parsapour, M.; Taghirad, H.D. Kernel-Based Sliding Mode Control for Visual Servoing System. *IET Comput. Vis.* **2015**, *9*, 309–320. [CrossRef]
25. Liu, H.; Zhu, W.; Dong, H.; Ke, Y. Hybrid Visual Servoing for Rivet-in-Hole Insertion Based on Super-Twisting Sliding Mode Control. *Int. J. Control Autom. Syst.* **2020**, *18*, 2145–2156. [CrossRef]
26. Miranda-Moya, A.; Castañeda, H.; Gordillo, J.L.; Wang, H. IBVS Based on Adaptive Sliding Mode Control for a Quadrotor Target Tracking under Perturbations. *Mechatronics* **2022**, *88*, 102909. [CrossRef]
27. Yuksel, T. Intelligent Sliding Surface Design Methods Applied to an IBVS System for Robot Manipulators. In *Applications from Engineering with MATLAB Concepts*; Valdman, J., Ed.; IntechOpen: Rijeka, Croatia, 2016.
28. Can, M.S.; Ercan, H. Real-Time Tuning of PID Controller Based on Optimization Algorithms for a Quadrotor. *Aircr. Eng. Aerosp. Technol.* **2022**, *94*, 418–430. [CrossRef]
29. Palm, R.; Driankov, D.; Hellendoorn, H. *Model Based Fuzzy Control Fuzzy Gain Schedulers and Sliding Mode Fuzzy Controllers*; Springer: New York, NY, USA, 1996.
30. Bartoszewicz, A.; Nowacka-Leverton, A. *Time-Varying Sliding Modes for Second and Third Order Systems*; Springer: Berlin/Heidelberg, Germany, 2009; ISBN 9783540922162.
31. Tokat, S.; Fadali, M.S.; Eray, O. A Classification and Overview of Sliding Mode Controller Sliding Surface Design Methods. In *Recent Advances in Sliding Modes: From Control to Intelligent Mechatronics*; Springer: Berlin/Heidelberg, Germany, 2015.
32. Eker, I. Sliding Mode Control with PID Sliding Surface and Experimental Application to an Electromechanical Plant. *ISA Trans.* **2006**, *45*, 109–118. [CrossRef] [PubMed]
33. Kowalska, O.; Kamiński, M.; Szabat, K. Implementation of a Sliding-Mode Controller with an Integral Function and Fuzzy Gain Value for the Electrical Drive with an Elastic Joint. *IEEE Trans. Ind. Electron.* **2010**, *57*, 1309–1317. [CrossRef]

34. Kim, J.-J.; Lee, J.-J.; Park, K.-B.; Youn, M.-J. Design of New Time-Varying Sliding Surface for Robot Manipulator Using Variable Structure Controller. *Electron. Lett.* **1993**, *29*, 195–196. [CrossRef]
35. Corke, P.I. *Robotics, Vision and Control: Fundamental Algorithms in MATLAB*, 2nd ed.; Springer: Berlin/Heidelberg, Germany, 2017.
36. Pounds, P.E.I. *Design, Construction and Control of a Large Quadrotor Micro Air Vehicle*; The Australian National University: Canberra, Australia, 2007.
37. Ercan, H.; Ulucan, H.; Can, M.S. Investigation of Wind Effect on Different Quadrotors. *Aircr. Eng. Aerosp. Technol.* **2022**, *94*, 1275–1288. [CrossRef]
38. Colley, S.J. *Vector Calculus*; Pearson: London, UK, 2012.