

Article

Image-Based Visual Servoing for Three Degree-of-Freedom Robotic Arm with Actuator Faults

Jiashuai Li, Xiuyan Peng, Bing Li *, Mingze Li and Jiawei Wu

College of Intelligent Systems Science and Engineering, Harbin Engineering University, Nantong Street, Harbin 150001, China; lijashuai@hrbeu.edu.cn (J.L.); pengxiuyan@hrbeu.edu.cn (X.P.); limingze123@hrbeu.edu.cn (M.L.); wujiawei@hrbeu.edu.cn (J.W.)

* Correspondence: libing265@hrbeu.edu.cn

Abstract: This study presents a novel image-based visual servoing fault-tolerant control strategy aimed at ensuring the successful completion of visual servoing tasks despite the presence of robotic arm actuator faults. Initially, a depth-independent image-based visual servoing model is established to mitigate the effects of inaccurate camera parameters and missing depth information on the system. Additionally, a robotic arm dynamic model is constructed, which simultaneously considers both multiplicative and additive actuator faults. Subsequently, model uncertainties, unknown disturbances, and coupled actuator faults are consolidated as centralized uncertainties, and an iterative learning fault observer is designed to estimate them. Based on this, suitable sliding surfaces and control laws are developed within the super-twisting sliding mode visual servo controller to rapidly reduce control deviation to near zero and circumvent the chattering phenomenon typically observed in traditional sliding mode control. Finally, through comparative simulation between different control strategies, the proposed method is shown to effectively counteract the effect of actuator faults and exhibit robust performance.

Keywords: image-based visual servoing; fault-tolerant control; iterative learning; sliding mode control; robotic arm



Citation: Li, J.; Peng, X.; Li, B.; Li, M.; Wu, J. Image-Based Visual Servoing for Three Degree-of-Freedom Robotic Arm with Actuator Faults. *Actuators* **2024**, *13*, 223. <https://doi.org/10.3390/act13060223>

Received: 26 April 2024

Revised: 28 May 2024

Accepted: 5 June 2024

Published: 13 June 2024



Copyright: © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Visual servoing is a vision-based robotic control method that allows the robot to interact with its environment by capturing images using a camera and extracting image feature information to achieve closed-loop control. Visual servoing technology enables robots to fully perceive their task environment and has been widely researched and applied in areas such as target grasping and tracking [1–3]. Depending on the number and types of cameras employed, visual servoing can be categorized into monocular visual servoing [4], binocular visual servoing [5], and multi-camera visual servoing systems [6]. According to the different image features used by robots, visual servoing can be divided into position-based visual servoing (PBVS) [7–9], image-based visual servoing (IBVS) [10–12], and hybrid visual servoing (HVS) [13–15]. PBVS employs 3-DOF position information to achieve visual servo control of a robot. In contrast, IBVS utilizes 2-DOF coordinates of the image feature points to achieve closed-loop control. HVS, on the other hand, integrates feedback information from both 2-DOF and 3-DOF. Compared to PBVS, IBVS can circumvent the issue of the target object leaving the camera's field of view. When contrasted with HVS, IBVS can avoid complex calculations, thereby enhancing the control system's real-time performance. Moreover, IBVS exhibits greater tolerance for robot modelling errors, camera calibration errors, and image noise. As a result, IBVS demonstrates superior control performance in non-structured control scenarios [16]. This study proposes a novel fault-tolerant control approach for IBVS.

In classical IBVS controllers, accurate camera intrinsic and extrinsic matrices are crucial for achieving precise control. However, obtaining accurate calibration parameters is often infeasible in most scenarios. To overcome this limitation, the uncalibrated visual servo

technique has been proposed, which does not require accurate calibration parameters and uses image feature motion during movement to estimate the robot's motion in real-time, providing good control performance in unknown and complex environments [17,18]. Numerous studies have been conducted to enhance the servoing efficiency and robustness of uncalibrated visual servoing [19–25]. Norouzi-Ghazbi et al. proposed an IBVS method based on model predictive control for visual servoing of continuum robots constrained by visual and actuator constraints, achieving endpoint control in the presence of constraints and improving the effectiveness and robustness of the system [19]. A constraint visual servoing method based on a sliding mode observer and model predictive controller was proposed in [20], where the sliding mode observer was designed to observe model uncertainty and joint velocities of the manipulator, and the model predictive controller converted the IBVS task into a nonlinear optimization problem and fully considered torque and visibility constraints, demonstrating the control performance of the proposed method in 2-DOF and 6-DOF IBVS systems. A homography-based unmanned aerial vehicle (UAV) visual servoing method was proposed in [21], where a saturation model-generated page control input was introduced into the robust controller to ensure that image feature points remained in the camera's region of interest, and the asymptotic stability of the closed-loop system was proven. The authors of [22,23] investigated the estimation problem of the image interaction matrix in IBVS. In [22], a robust Kalman filter visual servoing controller based on long short-term memory was proposed, which utilized the long short-term memory state estimation of the image interaction matrix and improved the initialization accuracy of the Jacobian matrix. In [23], the bidirectional extreme learning machine algorithm was adopted to approximate the image interaction matrix, exhibiting strong robustness to camera calibration errors and image noise. In [24,25], the authors adopted the reinforcement learning algorithm to adaptively tune visual servoing gains, effectively improving visual servoing efficiency compared to the traditional method with a fixed gain. Aghili proposed a novel hierarchical control visual servoing method based on adaptive Kalman filtering, considering the scenario where the visual system is temporarily occluded and completely fails. This method addresses both physical and operational constraints, achieving the capture of free-floating objects [26].

Although the aforementioned studies have made progress in achieving stable control of visual servoing, they all assume that the robot maintains normal operation during movement and overlook the possibility of robot failures. Due to long-term and repetitive work, the robot's actuators may experience degradation or even damage. To enhance the reliability and stability of visual servoing, it is necessary to consider the impact of actuator faults on visual servoing performance [27]. Fault-tolerant control can be broadly classified into two categories, which are active fault-tolerant control and passive fault-tolerant control. Active fault-tolerant control compensates for the robot's faults by detecting the fault information and providing the corresponding control signal through a fault-tolerant controller, making it suitable for high-precision control scenarios [28]. In contrast, passive fault-tolerant control considers the entire system and introduces fault-tolerance mechanisms to bring reliability [29]. Vision-based control of robotic arms is often applied in scenarios such as sorting and target capturing, which demands high precision of the control system. Therefore, the active fault-tolerant control method is more suitable for addressing the visual servoing problem of a robotic arm with actuator faults. Regarding fault detection methods for time-varying actuator faults, refs. [30,31] adopted the adaptive Radial Basis Function (RBF) neural network-based state observer to estimate the fault information. Le and Kang unified model uncertainties, disturbance torques, and fault information as centralized uncertainties and established an extended state observer to estimate the uncertainties [32]. A high-order sliding mode observer that improves convergence speed by adding a linear term was proposed in [33] to estimate joint fault status, and Lyapunov theory was used to demonstrate the system's finite-time stability. In designing fault-tolerant controllers for robotic arms, fuzzy logic controller [34], neural network controller [35], and sliding mode controller [36–39] have been adopted to input appropriate joint torques. Among them, the sliding mode control method has been widely studied due to its advantages of a simple design process and strong robustness. To overcome

the disadvantages of the traditional sliding mode controller, such as low convergence speed and control signal oscillation, terminal sliding mode, fast terminal sliding mode, and non-singular fast terminal sliding mode methods have been developed [36,37]. In sliding mode control, problems such as load disturbance and slow system response due to discontinuity of control input often occur at the beginning of the system. To address these issues, refs. [38,39] introduced the super-twisting method to provide additional control force, improving the system's response speed and control accuracy.

Guided by existing achievements, this study proposes an active fault-tolerant control method for a vision-based robotic arm, which is capable of handling actuator faults. The main contributions of this study are as follows:

- (1) Compared to [30], this study simultaneously considers the impact of both multiplicative and additive actuator faults on the system and reconstructs model uncertainties, load disturbances, and actuator faults as centralized uncertainties. The iterative learning fault observer (ILFO) is proposed to accurately and quickly estimate time-varying uncertainties. The dynamic relationship between the image feature point and the robotic arm is established, and super-twisting sliding mode visual servo controller (STSMVSC) with suitable sliding surfaces and control laws is designed to provide appropriate input torques to complete the visual servoing trajectory tracking task. Moreover, the stability analysis of the proposed observer and controller is provided using the Lyapunov theory.
- (2) In the simulation, the effectiveness and robustness of the proposed fault-tolerant control method are demonstrated under different coupled actuator fault severities. Compared to the fault-tolerant control method based on the extended state observer and STSMVSC, the proposed method can more quickly and accurately approach centralized uncertainties. Compared to the fault-tolerant control method based on ILFO and a traditional sliding mode visual servo controller, the proposed method can effectively reduce the chattering phenomenon and provide superior tracking performance.

The subsequent structure of the study is as follows: Section 1 presents the kinematic and dynamic models of visual servoing with simultaneous multiplicative and additive actuator faults. Section 2 introduces the iterative learning observer and the super-twisting sliding mode controller, along with the analysis of stability. Comparative simulations are presented in Section 3 to verify the effectiveness, generalization capability, and robustness of the proposed method. Finally, Section 4 summarizes the work presented in this study.

2. Problem Formulation

An IBVS system that experiences actuator faults can be described in Figure 1. The camera perceives the pose of image feature points in real-time and feeds them back to the controller. Based on the current pose of the feature points and the desired pose, the controller generates control signals that drive the motion of the robotic arm, allowing the control error to converge to an acceptable range. In this section, an IBVS model is developed, along with a dynamics model for a robotic arm experiencing actuator faults.

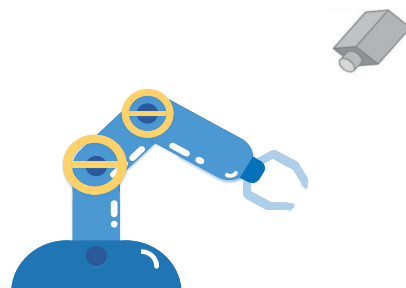


Figure 1. Schematic of visual servoing for a robotic arm with actuator faults.

2.1. Image-Based Visual Servoing

In IBVS, the control error occurs in the 2D image plane, while the 2D coordinates of feature points are obtained from their Cartesian coordinates through a perspective projection model. The image coordinates of the feature point $s_i = \begin{pmatrix} m_i & n_i \end{pmatrix}^T$ can be represented as follows:

$$s_i = \begin{pmatrix} m_i \\ n_i \end{pmatrix} = \frac{1}{D_z} \begin{pmatrix} u_1^T \\ u_2^T \end{pmatrix} T \begin{pmatrix} p_i \\ 1 \end{pmatrix}, \quad (1)$$

where m_i and n_i represent the coordinates of the feature point projected onto the m - and n -axes of the image plane, respectively; D_z represents the depth value of the feature points; u_i^T denotes the i th row vector of the unknown camera intrinsic and extrinsic matrix $U \in \mathbb{R}^{3 \times 4}$; $T \in \mathbb{R}^{4 \times 4}$ denotes the homogeneous transformation matrix, which is up to the forward kinematics of the robotic arm. T is given by $T = \begin{pmatrix} R_T & P_T \\ 0_{1 \times 3} & 1 \end{pmatrix}$, where $R_T \in \mathbb{R}^{3 \times 3}$ and $P_T \in \mathbb{R}^{3 \times 1}$ represent the rotation matrix and translation matrix, respectively; $p_i = \begin{pmatrix} X_i & Y_i & Z_i \end{pmatrix}^T$ denotes the Cartesian coordinates of the feature point [40]. By taking the time derivative of both sides of Equation (1), the dynamic relationship between the velocity of the feature point and the robotic arm can be obtained:

$$\begin{aligned} \dot{s}_i &= \frac{1}{D_z^2} \left(\begin{pmatrix} u_1^T \\ u_2^T \end{pmatrix} T \begin{pmatrix} \dot{p}_i \\ 1 \end{pmatrix} \dot{v} D_z - \dot{D}_z \begin{pmatrix} u_1^T \\ u_2^T \end{pmatrix} T \begin{pmatrix} p_i \\ 1 \end{pmatrix} \right) \\ &= \frac{1}{D_z} \begin{pmatrix} u_1^T \\ u_2^T \end{pmatrix} T \begin{pmatrix} \dot{p}_i \\ 1 \end{pmatrix} \dot{v} - \frac{\dot{D}_z}{D_z} \begin{pmatrix} u_1^T \\ u_2^T \end{pmatrix} T \begin{pmatrix} p_i \\ 1 \end{pmatrix}, \end{aligned} \quad (2)$$

where $\dot{v}$ denotes joint velocities of the robotic arm. The depth D_z can be represented as follows:

$$D_z = u_3^T T \begin{pmatrix} p_i \\ 1 \end{pmatrix}. \quad (3)$$

Taking the time derivative of Equation (3) yields

$$\dot{D}_z = u_3^T T \begin{pmatrix} \dot{p}_i \\ 1 \end{pmatrix} \dot{v}. \quad (4)$$

By substituting Equation (4) into Equation (2), Equation (2) can be simplified to

$$\begin{aligned} \dot{s}_i &= \frac{1}{D_z} \begin{pmatrix} m_i - u_1^T \\ n_i - u_2^T \end{pmatrix} T \begin{pmatrix} \dot{p}_i \\ 1 \end{pmatrix} \dot{v} \\ &= \frac{1}{D_z} J \dot{v}, \end{aligned} \quad (5)$$

where matrix J represents the depth-independent image interaction matrix that does not contain depth information.

The depth-independent image interaction matrix J and the depth of feature point D_z are nonlinear. However, they can be represented linearly with regressor matrices and unknown parameter vectors by the following property.

Property 1. For any vector χ , the products $J\chi$ and $D_z\chi$ can be parameterized in a linear form as

$$J\chi = A(\chi, v, s)\vartheta, \quad (6)$$

$$D_z\chi = B(\chi, s)\vartheta, \quad (7)$$

where $A(\chi, v, s)$ and $B(\chi, s)$ are the regressor matrices. ϑ is the corresponding parameter vector determined by the products of the camera parameters and the robot kinematic parameters.

2.2. Faulty Robotic Arm Dynamic Model

The dynamic equation of a rigid robotic arm with n degrees of freedom is typically described as follows:

$$W(v)\ddot{v} + C(v, \dot{v})\dot{v} + K(v) + F_r(v, \dot{v}) = \tau_r - \tau_{dt}, \quad (8)$$

where $v \in \mathbb{R}^n$, $\dot{v} \in \mathbb{R}^n$, $\ddot{v} \in \mathbb{R}^n$ represent the joint angle, angular velocity, and angular acceleration of the robotic arm, $W(v) \in \mathbb{R}^{n \times n}$ denotes the symmetric and positive definite inertia matrix, $C(v, \dot{v}) \in \mathbb{R}^{n \times n}$ denotes the skew-symmetric matrix of the Coriolis and centripetal forces, $K(v) \in \mathbb{R}^{n \times 1}$ denotes the gravity matrix, $F_r(v, \dot{v}) \in \mathbb{R}^{n \times 1}$ denotes the friction matrix, $\tau_r \in \mathbb{R}^{n \times 1}$ denotes the joints torque, and $\tau_{dt} \in \mathbb{R}^{n \times 1}$ denotes the disturbance matrix. When there exist uncertainties in the dynamics of the robotic arm, Equation (8) is expressed as follows:

$$\begin{aligned} & (W_n(v) + \Delta W(v))\ddot{v} + (C_n(v, \dot{v}) + \Delta C(v, \dot{v}))\dot{v} \\ & + (K_n(v) + \Delta K(v)) + F_r(v, \dot{v}) = \tau_r - \tau_{dt}, \end{aligned} \quad (9)$$

where $W_n(v)$, $C_n(v, \dot{v})$, $K_n(v)$ denotes the known dynamics, $\Delta W(v)$, $\Delta C(v, \dot{v})$, $\Delta K(v)$ denotes the unknown dynamics. Harsh environments and long-term operations can cause robotic arms to experience actuator multiplicative and additive faults. However, this paper does not focus on a specific type of actuator fault in robotic arms but adopts a more general actuator fault model. The paper considers the coupled actuator faults that include both multiplicative and additive faults, and models the fault torque as [41]

$$\tau_r^* = (I_n - \omega)\tau_r + \tau_a, \quad (10)$$

where $\tau_r^* \in \mathbb{R}^{n \times 1}$ denotes the degraded torque, $\omega \in \text{diag}(\omega_1, \omega_2, \dots, \omega_n)$ denotes the remaining control factor, and $\tau_a \in \mathbb{R}^{n \times 1}$ denotes the additive fault torque. By substituting Equation (10) into Equation (9), the dynamic equation of robot manipulation with actuator fault can be expressed as follows:

$$\begin{aligned} & (W_n(v) + \Delta W(v))\ddot{v} + (C_n(v, \dot{v}) + \Delta C(v, \dot{v}))\dot{v} \\ & + (K_n(v) + \Delta K(v)) + F_r(v, \dot{v}) = (I_n - \omega)\tau_r + \tau_a - \tau_{dt}. \end{aligned} \quad (11)$$

The above equation can be simplified as follows:

$$\begin{aligned} & W_n(v)\ddot{v} + C_n(v, \dot{v})\dot{v} + K_n(v) \\ & = \tau_r - \omega\tau_r + \tau_a - \tau_{dt} + \Delta W(v)\ddot{v} - \Delta C(v, \dot{v})\dot{v} \\ & \quad - \Delta K(v) - F_r(v, \dot{v}) \\ & = \tau_r - \Phi(v, \dot{v}), \end{aligned} \quad (12)$$

where $\Phi(v, \dot{v})$ denotes the concentrated uncertainty term containing model uncertainty, frictional torque, and fault torque. Then, the dynamics of the robot manipulation can be expressed as follows:

$$\ddot{v} = W_n^{-1}(v)(\tau_r - \Phi(v, \dot{v}) - C_n(v, \dot{v})\dot{v} - K_n(v)). \quad (13)$$

3. Fault-Tolerant Visual Servo Control

3.1. Iterative Learning Observer

To facilitate the design of state observers and fault-tolerant visual servo controllers, the following state variables are defined:

$$x(t) = \begin{bmatrix} x_1(t) \\ x_2(t) \end{bmatrix} = \begin{bmatrix} v(t) \\ \dot{v}(t) \end{bmatrix}, \quad (14)$$

$$y(t) = Cx(t), \quad (15)$$

$$u(t) = \tau_r(t). \quad (16)$$

Then, Equation (13) can be rewritten as the following state-space equation:

$$\begin{cases} \dot{x}(t) = Ax(t) + B(x(t), u(t)) + D(x(t)) + E(x(t))\Phi(t) \\ y(t) = Cx(t), \end{cases} \quad (17)$$

$$\text{where } A = \begin{bmatrix} 0_n I_n \\ 0_n 0_n \end{bmatrix},$$

$$B(x(t), u(t)) = \begin{bmatrix} 0_n \\ W_n^{-1}(x_1(t))u(t) \end{bmatrix},$$

$$C = I_{2n},$$

$$D(x(t)) = \begin{bmatrix} 0_n \\ W_n^{-1}(x_1(t))(-C_n(v, \dot{v})\dot{v} - K_n(v)) \end{bmatrix},$$

$$E(x(t)) = \begin{bmatrix} 0_n \\ -W_n^{-1}(x_1(t)) \end{bmatrix},$$

Where 0_n represents an $n \times n$ zero matrix, where all elements are zeros. I_n represents an $n \times n$ identity matrix, which has ones on the diagonal and zeros elsewhere, I_{2n} represents an $2n \times 2n$ identity matrix, which has ones on the diagonal and zeros elsewhere.

To accurately estimate the aggregated uncertainties, design the following iterative learning fault observer:

$$\begin{cases} \hat{x}(t) = A\hat{x}(t) + B(\hat{x}(t), u(t)) + D(\hat{x}(t)) + G(y(t) - \hat{y}(t)) + E\zeta(t) \\ \zeta(t) = r_1\zeta(t - \kappa) + r_2[y(t) - \hat{y}(t)]\hat{y}(t) = C\hat{x}(t) \end{cases}, \quad (18)$$

where $\hat{x}(t)$ denotes the state estimation, $\hat{y}(t)$ denotes the output estimation, κ denotes the sampling time, $\zeta(t)$ denotes the iterative observer input at time t , $\zeta(t - \kappa)$ denotes the iterative observer input from the previous sampling period, and G, r_1, r_2 are the observer gain matrices.

Remark 1. In practical applications, the velocity state of a robotic arm often cannot be directly measured; typically, only joint angle information is available. In this study, the ILFO is designed to accurately and quickly estimate unmeasurable velocity states, as well as other time-varying uncertainties and actuator faults. By employing an iterative learning mechanism, this observer can continuously update its estimates in a continuous-time model, thereby improving the dynamic estimation accuracy and response speed of the system.

Remark 2. The system dynamics are inherently continuous, as described by the state estimation $\hat{x}(t)$ and the output estimation $\hat{y}(t)$ in Equation (18). These continuous-time expressions form the core of the system's behavior. The variable κ denotes this sampling interval, representing the discrete time steps at which the iterative learning updates occur. $\zeta(t - \kappa)$ refers to the iterative observer input from the previous sampling period. This discrete update mechanism is integrated into the continuous-time model to leverage the benefits of both continuous-time system dynamics and discrete-time learning adjustments. Thus, the update of $\zeta(t)$ occurs at discrete intervals, while the overall system operates in continuous time.

Define the estimation error as follows:

$$\begin{cases} e_x(t) = x(t) - \hat{x}(t) \\ e_y(t) = y(t) - \hat{y}(t). \\ \tilde{\zeta}(t) = \Phi(t) - \zeta(t) \end{cases} \quad (19)$$

Substituting Equations (17) and (18) into Equation (19), the dynamic equation for the state observation error can be expressed as follows:

$$\begin{aligned}
 \dot{e}_x(t) &= \dot{x}(t) - \dot{\hat{x}}(t) \\
 &= A(x(t) - \hat{x}(t)) \\
 &\quad + (B(x(t), u(t)) - B(\hat{x}(t), u(t))) \\
 &\quad + (D(x(t)) - D(\hat{x}(t))) + E(x(t))\Phi(t) \\
 &\quad - G(y(t) - \hat{y}(t)) - E(x(t))\zeta(t) \\
 &= (A - GC)e_x(t) \\
 &\quad + (B(x(t), u(t)) - B(\hat{x}(t), u(t))) \\
 &\quad + (D(x(t)) - D(\hat{x}(t))) + E(x(t))(\Phi(t) - \zeta(t)) \\
 &= (A - GC)e_x(t) \\
 &\quad + (B(x(t), u(t)) - B(\hat{x}(t), u(t))) \\
 &\quad + (D(x(t)) - D(\hat{x}(t))) + E(x(t))\tilde{\zeta}(t).
 \end{aligned} \tag{20}$$

First, the assumption and theorem for the proposed iterative learning sliding mode observer are as follows:

Assumption 1. Assume that $B(x(t), u(t))$ and $D(x(t))$ are bounded, and there exists Lipschitz constants o_1, o_2 such that the following inequality holds:

$$\begin{aligned}
 \|B(x(t), u(t)) - B(\hat{x}(t), u(t))\| &\leq o_1 \|x(t) - \hat{x}(t)\|, \\
 \|D(x(t)) - D(\hat{x}(t))\| &\leq o_2 \|x(t) - \hat{x}(t)\|.
 \end{aligned} \tag{21}$$

Theorem 1. For the robot arm dynamics Equation (17) and the iterative learning observer Equation (18), there exist symmetric matrices $P \in \mathbb{R}^{2n \times 2n}$, $Q \in \mathbb{R}^{2n \times 2n}$ that satisfy the following equation:

$$(A - GC)^T P + P(A - GC) + 2o_1 P + 2o_2 P = -Q, \tag{22}$$

and satisfying Assumption 1 can make the errors of the state and concentrated uncertainty estimation consistently asymptotically stable.

Proof. Define the Lyapunov equation as follows:

$$V = e_x^T(t) P e_x(t) + \int_{t-\kappa}^t \tilde{\zeta}^T(s) \tilde{\zeta}(s) ds. \tag{23}$$

Taking the derivative of the above equation and substituting Equation (20) yields

$$\begin{aligned}
 \dot{V} &= e_x^T(t) \left((A - GC)^T P + P(A - GC) \right) e_x(t) \\
 &\quad + 2e_x^T(t) P [D(x(t)) - D(\hat{x}(t))] \\
 &\quad + 2e_x^T(t) P [B(x(t), u(t)) - B(\hat{x}(t), u(t))] \\
 &\quad + 2e_x^T(t) P E \tilde{\zeta}(t) \\
 &\quad + \tilde{\zeta}^T(t) \tilde{\zeta}(t) - \tilde{\zeta}^T(t - \kappa) \tilde{\zeta}(t - \kappa).
 \end{aligned} \tag{24}$$

Based on Equation (21), it can be obtained that

$$\begin{aligned}
 \dot{V} &\leq e_x^T(t) \left((A - GC)^T P + P(A - GC) \right. \\
 &\quad \left. + 2Po_1 + 2Po_2 \right) e_x(t) \\
 &\quad + 2e_x^T(t) P E \tilde{\zeta}(t) + \tilde{\zeta}^T(t) \tilde{\zeta}(t) \\
 &\quad - \tilde{\zeta}^T(t - \kappa) \tilde{\zeta}(t - \kappa).
 \end{aligned} \tag{25}$$

Construct the auxiliary variable as follows:

$$\zeta_d(t) = \Phi(t) - r_1\Phi(t - \kappa). \quad (26)$$

Then,

$$\tilde{\zeta}(t) = r_1\tilde{\zeta}(t - \kappa) - r_2C\hat{x}(t) + \zeta_d(t), \quad (27)$$

$$\begin{aligned} \dot{V} \leq & e_x^T(t) \left((A - GC)^T P + P(A - GC) \right) e_x(t) \\ & + 2Pe_1 + 2Pe_2 \\ & + 2e_x^T(t)PEr_1\tilde{\zeta}(t - \kappa) - 2e_x^T(t)PEr_2C\hat{x}(t) \\ & + 2e_x^T(t)PE\zeta_d(t) \\ & + \tilde{\zeta}^T(t)\tilde{\zeta}(t) - \tilde{\zeta}^T(t - \kappa)\tilde{\zeta}(t - \kappa). \end{aligned} \quad (28)$$

By introducing positive constants α, β that satisfy $\alpha - \beta = 1$, the above equation can be expressed as follows:

$$\begin{aligned} \dot{V} \leq & e_x^T(t) \left((A - GC)^T P + P(A - GC) \right) e_x(t) \\ & + 2Pe_1 + 2Pe_2 \\ & + 2e_x^T(t)PEr_1\tilde{\zeta}(t - \kappa) - 2e_x^T(t)PEr_2C\hat{x}(t) \\ & + \alpha\tilde{\zeta}^T(t)\tilde{\zeta}(t) - \beta\tilde{\zeta}^T(t)\tilde{\zeta}(t) - \tilde{\zeta}^T(t - \kappa)\tilde{\zeta}(t - \kappa) \\ & + 2e_x^T(t)PE\zeta_d(t). \end{aligned} \quad (29)$$

Substituting Equation (27) into the above equation yields

$$\begin{aligned} \dot{V} \leq & e_x^T(t) \left((A - GC)^T P + P(A - GC) \right) e_x(t) \\ & + 2Pe_1 + 2Pe_2 \\ & + 2e_x^T(t)PEr_1\tilde{\zeta}(t - \kappa) - 2e_x^T(t)PEr_2C\hat{x}(t) \\ & + 2e_x^T(t)PE\zeta_d(t) \\ & - \beta\tilde{\zeta}^T(t)\tilde{\zeta}(t) - \tilde{\zeta}^T(t - \kappa)\tilde{\zeta}(t - \kappa) \\ & + \alpha\tilde{\zeta}_d^T(t)\zeta_d(t) + 2\alpha\tilde{\zeta}^T(t - \kappa)r_1^T\zeta_d(t) \\ & - 2\alpha e_x^T(t)(r_2C)^T\zeta_d(t) \\ & + \alpha\tilde{\zeta}^T(t - \kappa)r_1^Tr_1\tilde{\zeta}(t - \kappa). \end{aligned} \quad (30)$$

If both Equation (22) and the following relationship are satisfied

$$PE = \alpha(r_2C)^T, \quad (31)$$

it yields

$$\begin{aligned} \dot{V} \leq & -\lambda_{\min}(Q)\|e_x(t)\|^2 - \beta\|\tilde{\zeta}(t)\|^2 \\ & - \tilde{\zeta}^T(t - \kappa)\tilde{\zeta}(t - \kappa) \\ & + \alpha\tilde{\zeta}^T(t - \kappa)r_1^Tr_1\tilde{\zeta}(t - \kappa) \\ & + 2\alpha\tilde{\zeta}^T(t - \kappa)r_1^T\zeta_d(t) \\ & + \alpha\tilde{\zeta}_d^2 + 2\mu\|P\|\|e_x(t)\|^2. \end{aligned} \quad (32)$$

Using the following inequality property:

$$\begin{aligned} 2\alpha\tilde{\zeta}^T(t - \kappa)r_1^T\zeta_d(t) \leq & l^2\tilde{\zeta}^T(t - \kappa)r_1^Tr_1\zeta_d(t) \\ & + \frac{\alpha^2}{l^2}\zeta_d^T(t)\zeta_d(t). \end{aligned} \quad (33)$$

Then,

$$\begin{aligned}\dot{V} \leq & -\lambda_{\min}(Q)\|e_x(t)\|^2 - \beta\|\tilde{\zeta}(t)\|^2 \\ & + \tilde{\zeta}^T(t-\kappa) \left[(\alpha + l^2)r_1^T r_1 - I \right] \tilde{\zeta}(t-\kappa) \\ & + \left(\alpha + \frac{\alpha^2}{l^2} \right) \tilde{\zeta}_d^2 + 2\mu\lambda_{\max}(P)\|e_x(t)\|^2.\end{aligned}\quad (34)$$

By defining $\nu = \lambda_{\min}(Q) - 2\mu\lambda_{\max}(P) > 0$, it yields

$$\begin{aligned}\dot{V} \leq & -\nu\|e_x(t)\|^2 - \beta\|\tilde{\zeta}(t)\|^2 + \left(\alpha + \frac{\alpha^2}{l^2} \right) \tilde{\zeta}_d^2 \\ & + \tilde{\zeta}^T(t-\kappa) \left[(\alpha + l^2)r_1^T r_1 - I \right] \tilde{\zeta}(t-\kappa).\end{aligned}\quad (35)$$

If condition $l > (\alpha + l^2)r_1^T r_1 > 0$ holds, then the iterative observer is uniformly ultimately stable, and the proof is complete. $\square$

3.2. Fault-Tolerant Sliding Mode Controller

The control system diagram is shown in Figure 2. The objective of this section is to design a fault-tolerant IBVS controller that drives a robotic arm to converge the image feature points to zero within a finite time, given the initial position and the desired trajectory of the image feature points, after receiving the estimated fault values from the observer. To overcome the drawbacks of traditional sliding mode control, such as control signal chattering and low control accuracy, a super-twisting sliding mode controller is designed in this section. In sliding mode control, when the system state moves along the sliding surface $\sigma = 0$, it indicates that the actual state variables have completely tracked the target state variables. Before designing the sliding surface, we first define the system state variables.

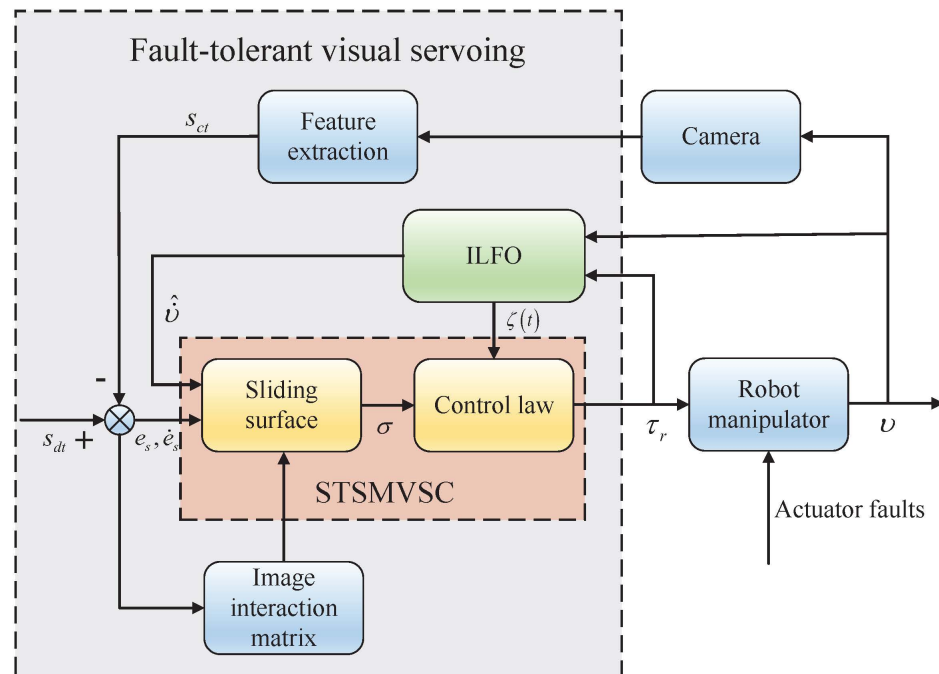


Figure 2. The control system diagram.

The tracking error is defined as

$$e_s = s_{dt} - s_{ct}, \quad (36)$$

where s_{dt} represents the desired trajectory of the feature points, s_{ct} represents the current coordinates of the feature points, as given by Equation (1). Based on Equations (1), (5) and (13), the first and second derivatives of the tracking error concerning time can be obtained

$$\dot{e}_s = \dot{s}_{dt} - \dot{s}_{ct}, \quad (37)$$

$$\ddot{e}_s = \ddot{s}_{dt} - \ddot{s}_{ct} = \Gamma \ddot{v} + \dot{\Gamma} \dot{v} - \ddot{s}_{ct}, \quad (38)$$

where $\Gamma = \frac{1}{D_z} J$.

Define the sliding surface as follows:

$$\sigma = \dot{e}_s + \delta e_s, \quad (39)$$

where δ is a positive constant that needs to be designed. After completing the design of the sliding surface, the next step is to design a suitable control law to make $\dot{\sigma} = 0$, which means making the system trajectory converge to the sliding surface. According to the state-space equation, Equations (17) and (37)–(39), the derivative of the sliding surface is expressed as

$$\begin{aligned} \dot{\sigma} &= \ddot{e}_s + \delta \dot{e}_s \\ &= \ddot{s}_{dt} - \ddot{s}_{ct} + \delta \dot{e}_s \\ &= \Gamma \ddot{v} + \dot{\Gamma} \dot{v} - \ddot{s}_{ct} + \delta \dot{e}_s \\ &= \Gamma W_n^{-1}(v)(\tau_r - \Phi(v, \dot{v}) - C_n(v, \dot{v})\dot{v} - K_n(v)) \\ &\quad + \dot{\Gamma} \dot{v} - \ddot{s}_{ct} + \delta \dot{e}_s \\ &= \Gamma W_n^{-1}(x_1(t))u - \Gamma W_n^{-1}(x_1(t))\Phi(v, \dot{v}) \\ &\quad - \Gamma W_n^{-1}(x_1(t))(C_n(v, \dot{v})\dot{v} + K_n(v)) \\ &\quad + \dot{\Gamma} \dot{v} - \ddot{s}_{ct} + \delta \dot{e}_s. \end{aligned} \quad (40)$$

In this study, we employ the use of the sign function, defined as follows:

$$\text{sign}(\sigma) = \begin{cases} 1 & \text{if } \sigma > 0 \\ 0 & \text{if } \sigma = 0 \\ -1 & \text{if } \sigma < 0 \end{cases}. \quad (41)$$

The visual servoing control law is designed as follows:

$$u = u_1 + u_2, \quad (42)$$

where u_1 is used to control the nominal term, expressed as

$$u_1 = C_n(v, \hat{v})\hat{v} + K_n(v) - W_n(v)\Gamma^{-1}(\dot{\Gamma}\hat{v} - \ddot{s}_{ct} + \delta \dot{e}_s), \quad (43)$$

where $\hat{v}$ represents the joint angular velocities estimated by ILFO. u_2 is used to control the effects caused by model uncertainties, unknown disturbances and faults, expressed as

$$\begin{aligned} u_2 &= W_n(v)\Gamma^{-1}(\mu_1 \|\sigma\|^{1/2} \text{sign}(\sigma) - \beta) + \zeta(t) \\ \dot{\beta} &= -\mu_2 \text{sign}(\sigma). \end{aligned} \quad (44)$$

Substituting Equations (42)–(44) into Equation (40), we obtain

$$\begin{aligned} \dot{\sigma} &= -\mu_1 \|\sigma\|^{1/2} \text{sign}(\sigma) + \beta + \lambda(t) \\ \dot{\beta} &= -\mu_2 \text{sign}(\sigma), \end{aligned} \quad (45)$$

where $\lambda(t) \leq \bar{\lambda}$ is a bounded element related to unknown disturbances, $\bar{\lambda}$ is the upper bound. When the robot model given by Equation (13) experiences a fault, by selecting an

appropriate control gain, the visual servoing tracking error can be made to converge to zero in a finite time when the sliding mode surface is determined by Equation (39) and the control rate is determined by Equations (42)–(44). A stability-proof method for a trajectory tracking controller for a robotic arm was proposed in [42]. Although it is not directly related to the problem discussed in this paper, it can be applied to the stability proof of the visual servoing controller proposed in this paper. Therefore, the stability proof of the controller is not separately provided in this paper.

4. Numerical Simulations

In this section, an IBVS trajectory tracking task is set up to validate the effectiveness, robustness, and generalization capability of the proposed method when the robotic arm experiences actuator faults. The 3-DOF robotic arm used in the simulation has parameters as shown in [43], and only the first three joints are used. The camera is fixed to the end-actuator of the robotic arm. The real values and roughly estimated values of the camera parameters are shown in Table 1, where α_m and α_n represent the scale factors along the m - and n - axes, respectively. The focal length of the camera is set to 0.005 m. To ensure system synchronization, the camera's image capture speed was set to 25 frames per second, and the control system's sampling time was set to 40 ms.

Table 1. Camera parameters.

Parameters	Real Value	Rough Estimated Value
m_0 (pixels)	640	600
n_0 (pixels)	512	500
α_m (pixels/m)	265,000	260,000
α_n (pixels/m)	266,500	260,000

In the simulation, friction forces and disturbance torques are taken into consideration and are set as follows:

$$F_r = \begin{bmatrix} F_{r1} \\ F_{r2} \\ F_{r3} \end{bmatrix} = \begin{bmatrix} 0.5\dot{v}_1 \\ 0.8\dot{v}_2 \\ 1.2\dot{v}_3 \end{bmatrix}, \quad (46)$$

$$\tau_{dt} = \begin{bmatrix} \tau_{dt1} \\ \tau_{dt2} \\ \tau_{dt3} \end{bmatrix} = \begin{bmatrix} 0.1 \sin(v_1) \\ 0.15 \sin(v_2) \\ 0.2 \sin(v_3) \end{bmatrix}.$$

To demonstrate that the proposed method is not limited to specific actuator fault parameters, two sets of robotic arm actuator fault parameters are given in the simulation to validate the stability and generalization capability of the proposed method.

Case 1 The simulation time is set to 50 s, with actuator faults occurring at the 20th, 30th and 40th seconds, respectively. The input torque with faults, as shown in Equation (10), is described as

$$\tau_r^* = \begin{bmatrix} \tau_{r1}^* \\ \tau_{r2}^* \\ \tau_{r3}^* \end{bmatrix} = \begin{bmatrix} 0.8\tau_{r1} - 1 \\ 0.7\tau_{r2} \\ \tau_{r3} - 1 \end{bmatrix}, t \in (20, 30, 40). \quad (47)$$

Case 2 The simulation time is set to 50 s, with a time-varying actuator fault persisting after the 20th second. The input torque with faults, as shown in Equation (10), is described as

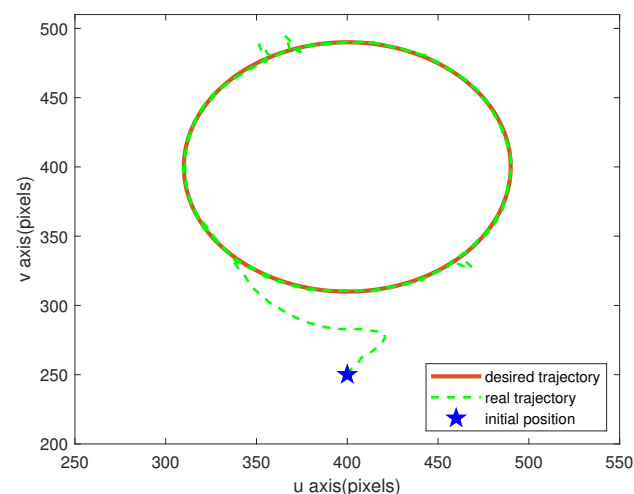
$$\tau_r^* = \begin{bmatrix} \tau_{r1}^* \\ \tau_{r2}^* \\ \tau_{r3}^* \end{bmatrix} = \begin{cases} \begin{bmatrix} \tau_{r1} \\ \tau_{r2} \\ \tau_{r3} \end{bmatrix}, t \in [0, 20) \\ \begin{bmatrix} 0.8\tau_{r1} \\ (0.75 - 0.1 \sin(t))\tau_{r2} \\ -0.5 \sin(t) \\ (0.8 + 0.1 \cos(t))\tau_{r3} - 1 \end{bmatrix}, t \in [20, 50]. \end{cases} \quad (48)$$

Furthermore, to demonstrate the effectiveness and superiority of the proposed method, the comparative simulation between the proposed method and the control strategies based on the extended state observer (ESO) [44] with super-twisting sliding mode visual servo controller (ESO-STSMVSC), and the control strategies based on the iterative learning fault observer with traditional sliding mode visual servo controller [38] (ILFO-SMVSC). The control objective is to make the image feature point track the desired trajectory.

$s_d = [400 + 90 \sin(0.3t), 400 + 90 \cos(0.3t)]^T$ from the initial coordinates $s_{initial} = [400, 260]^T$.

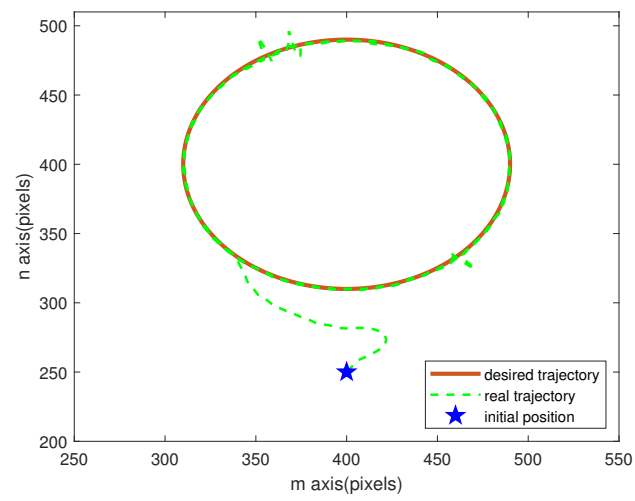
The parameters of ILFO corresponding to each degree of freedom are designed as $G = \begin{bmatrix} 20.38 & 12.53 \\ -8.3 & 22.51 \end{bmatrix}$, $r_1 = \text{diag}[0.85 \ 0.9]$, $r_2 = \text{diag}[20.5 \ 30]$. The parameters of STSMVSC are designed as $\mu_1 = 2.15$ and $\mu_2 = 2.5$.

The comparison results of trajectory tracking are illustrated in Figures 3–10. Figure 3 shows the trajectories of the image feature point under Case 1, with the blue pentagram representing the initial position of the feature point, and red and green curves representing the desired and real trajectories, respectively. It can be observed that, under all three control strategies, the feature points can track the desired trajectories from their initial positions. Figure 4 demonstrates the changes in tracking errors along the m - and n - axes. It is evident that, in the initial stage of tracking, compared to ILFO-SMVSC, both ILFO-STSMVSC, and ESO-STSMVSC enable faster convergence of image deviations. When actuator faults occur, compared to ESO-STSMVSC, both ILFO-STSMVSC and ILFO-SMVSC can more rapidly mitigate the impact of actuator faults on the system, driving the tracking error to converge into a small residual around zero. This indicates that the proposed method outperforms the other two control strategies concerning both error convergence speed and mitigating actuator fault disturbances.

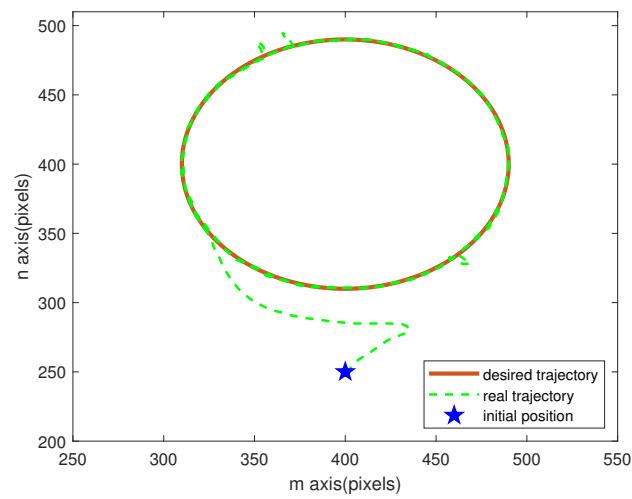


(a)

Figure 3. Cont.



(b)



(c)

Figure 3. Trajectories of the image feature point under Case 1. (a) Trajectory under ILFO-STSMVSC. (b) Trajectory under ESO-STSMVSC. (c) Trajectory under ILFO-SMVSC.

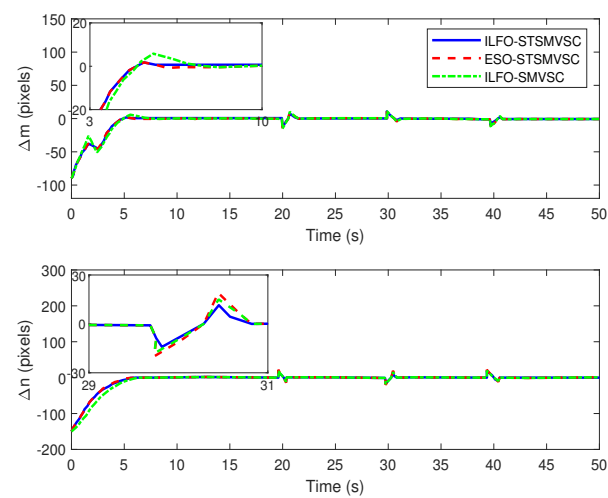


Figure 4. Tracking errors under Case 1.

Figure 5 presents a comparison of concentrated uncertainty estimation performance for different observers under Case 1. The ILFO can dynamically update estimation errors based on the sample time, offering higher estimation accuracy and faster convergence of estimation errors compared to ESO. Due to the presence of the iterative learning factor $\zeta(t)$, the observed values of the fault state can be adjusted in real-time during the observation process, enhancing the dynamic estimation accuracy of the observer, which helps improve the system's response speed and robustness. At the same time, by iteratively updating the historical data, the observer improves the estimation accuracy of the system state and output. The input $\zeta(t)$ in the iterative process is adjusted according to the iterative observer input $\zeta(t - \kappa)$ from the previous sampling period, gradually optimizing the observer's performance. Moreover, the observer can accurately estimate the state and output, enhancing the ability to detect system uncertainties and faults. Compared to ESO, ILFO requires relatively fewer parameters to be adjusted during implementation and can automatically optimize parameters through the iterative process, making the deployment and debugging of ILFO less challenging.

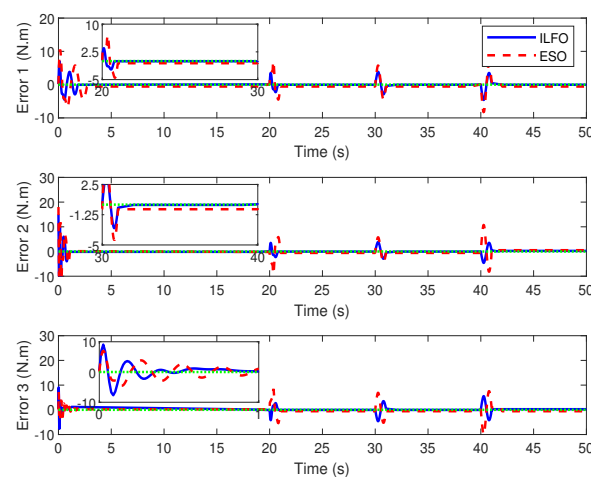


Figure 5. Estimation errors of concentration uncertainty under Case 1.

Figure 6 illustrates the comparison of input torques for the robotic arm under different controllers in Case 1. It can be observed that the control laws given by Equations (42)–(44) reduce the impact of switching gain on system performance, effectively mitigating the chattering phenomenon associated with the traditional sliding mode. Consequently, this leads to smoother control signals and an overall improvement in control quality. Figures 3–6 confirm that, under Case 1, the proposed controller demonstrates better control performance compared to ESO-STSMVSC and ILFO-SMVSC. The proposed fault observer and visual servo controller also exhibit superior performance.

Figures 7–10 presents the comparative simulation results under Case 2. Figures 7 and 8 indicate that, under time-varying actuator faults, the proposed control strategy offers superior tracking effects compared to ESO-STSMVSC and ILFO-SMVSC. In Figure 9, it can be observed that the estimation accuracy and speed of the proposed ILFO surpass those of ESO. Figure 10 displays the input torques under different control strategies, and similar to Case 1, the proposed controller effectively avoids the chattering phenomenon of the controller based on traditional sliding mode control.

Figures 11 and 12 show the velocities of the robotic arm's end-effector under the proposed control scheme in Case 1 and Case 2. It can be observed that the velocity of the end-effector changes smoothly. After being disturbed by a fault, the end-effector quickly moves to bring the system back to a stable tracking state. The above comparative simulations demonstrate the effectiveness, stability, and generalization capabilities of the proposed method under different time-varying actuator faults. The proposed method can effectively accomplish visual servo trajectory tracking tasks in the presence of actuator faults.

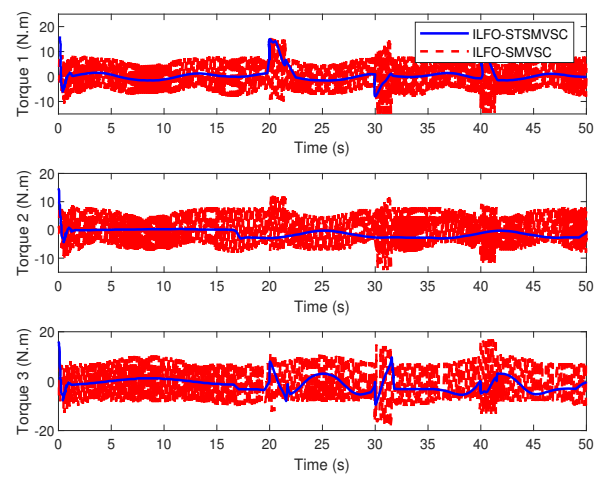
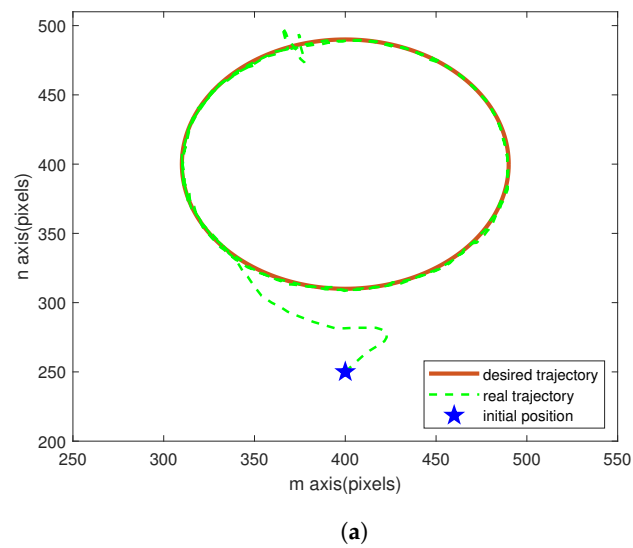
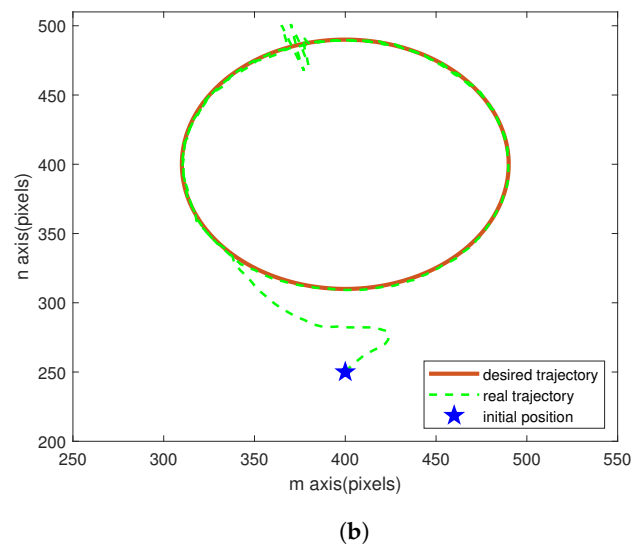


Figure 6. Input torque under Case 1.

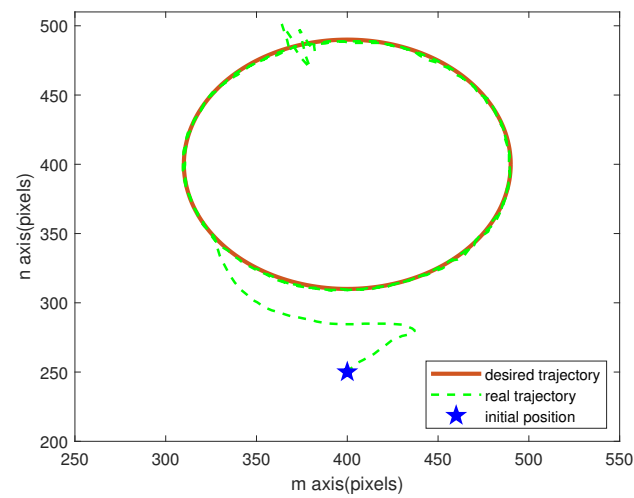


(a)



(b)

Figure 7. Cont.



(c)

Figure 7. Comparative trajectories of the image feature point under Case 2. (a) Trajectory under ILFO-STSMVSC. (b) Trajectory under ESO-STSMVSC. (c) Trajectory under ILFO-SMVSC.

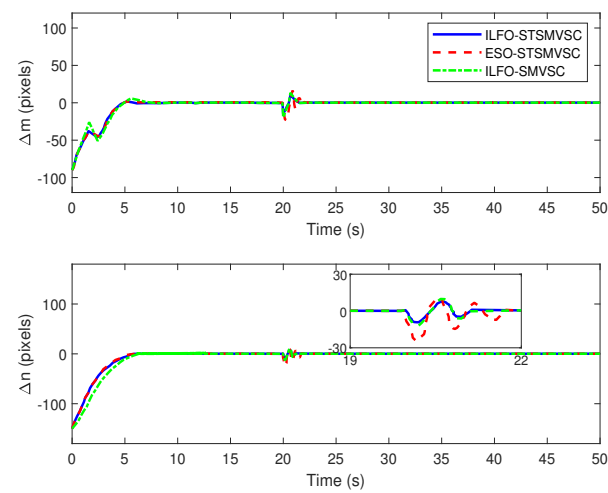


Figure 8. Tracking errors under Case 2.

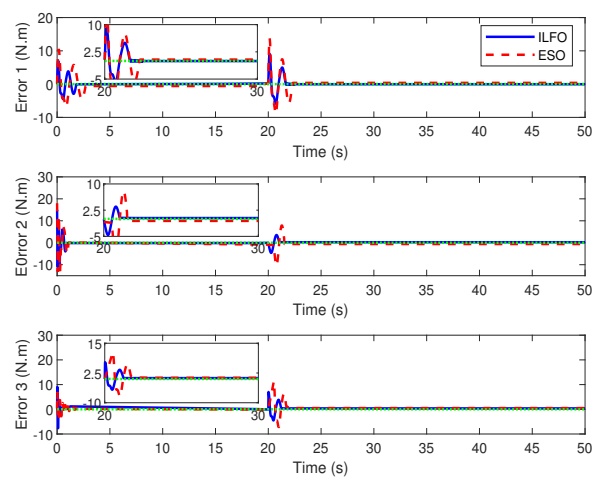


Figure 9. Estimation errors of concentration uncertainty under Case 2.

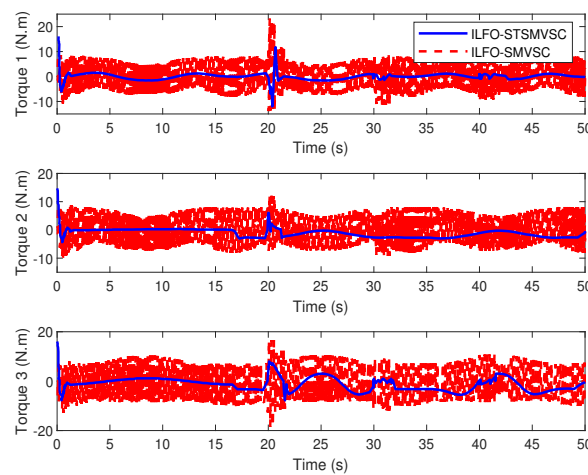


Figure 10. Input torque under Case 2.

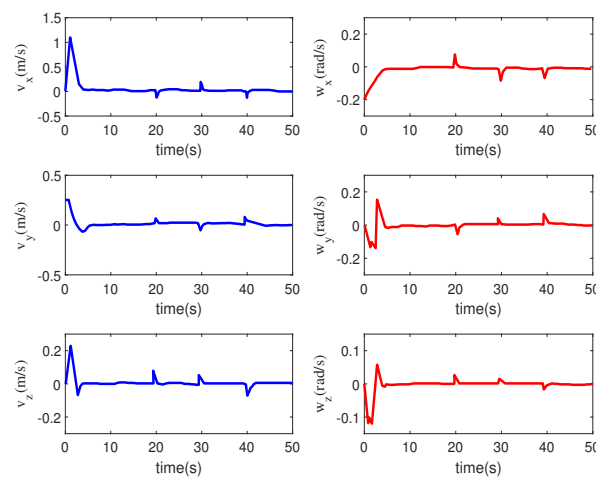


Figure 11. End-effector velocities of the robotic arm under Case 1.

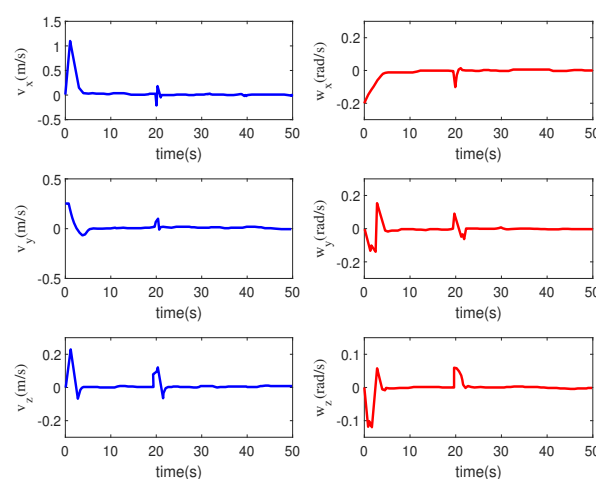


Figure 12. End-effector velocities of the robotic arm under Case 2.

5. Conclusions

This paper proposes an IBVS control strategy that utilizes an iterative learning observer and a super-twisting sliding mode controller for a robotic arm's visual servo system in the presence of actuator faults. In contrast to existing research, the impact of actuator

faults on robust visual servoing control is taken into account. The model uncertainty, load disturbance, and coupling actuator fault are reconstructed as centralized uncertainty, and the ILFO is designed to estimate it. The dynamic relationship between the image feature point and the robotic arm is established, and the STSMVSC is designed to mitigate the chattering phenomenon commonly observed in traditional sliding mode control and enhance convergence speed. Employing the Lyapunov theory, the stability of both the observer and controller is ensured. The stability and generalization capability of the proposed method under various actuator fault conditions are validated through simulation. The comparative simulation results indicate that, when faced with various actuator faults, the proposed method demonstrates superior servo efficiency and tracking accuracy compared to other IBVS approaches. Moreover, it effectively mitigates the impact of concentrated uncertainty on the system. In summary, the proposed method can accomplish uncalibrated visual servo trajectory tracking tasks even in the presence of actuator faults.

Author Contributions: Conceptualization, B.L.; Methodology, J.L., X.P., B.L., M.L. and J.W.; Formal Analysis, J.L., X.P., B.L., M.L. and J.W.; Data Curation, J.L.; Writing—Original Draft Preparation, J.L.; Writing—Review and Editing, J.L., X.P., B.L., M.L. and J.W.; Visualization, J.L.; Supervision, B.L.; Project Administration, B.L.; Funding Acquisition, B.L. All authors have read and agreed to the published version of the manuscript.

Funding: This study was funded by the Fundamental Strengthening Program Technical Field Fund, Grant/Award Number: 2021-JCJQ-JJ-0026.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The original contributions presented in the study are included in the article, further inquiries can be directed to the corresponding author.

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

References

1. Lagneau, R.; Krupa, A.; Marchal, M. Automatic shape control of deformable wires based on model-free visual servoing. *IEEE Robot. Autom. Lett.* **2020**, *5*, 5252–5259. [\[CrossRef\]](#)
2. Palmieri, P.; Troise, M.; Gaidano, M.; Melchiorre, M.; Mauro, S. Inflatable robotic manipulator for space debris mitigation by visual servoing. In Proceedings of the 2023 9th International Conference on Automation, Robotics and Applications (ICARA), Abu Dhabi, United Arab Emirates, 10–12 February 2023; pp. 175–179.
3. De Farias, C.; Adjigble, M.; Tamadazte, B.; Stolkin, R.; Marturi, N. Dual quaternion-based visual servoing for grasping moving objects. In Proceedings of the 2021 IEEE 17th International Conference on Automation Science and Engineering (CASE), Lyon, France, 23–27 August 2021; pp. 151–158.
4. Wang, N.; He, H.K. Extreme learning-based monocular visual servo of an unmanned surface vessel. *IEEE Trans. Ind. Inform.* **2021**, *17*, 5152–5163. [\[CrossRef\]](#)
5. Gao, H.; Shen, F.; Zhang, F.; Zhang, Z.T. A high precision and fast alignment method based on binocular vision. *Int. J. Precis. Eng. Manuf.* **2022**, *23*, 969–984. [\[CrossRef\]](#)
6. Dallej, T.; Gouttefarde, M.; Andreff, N.; Herve, P.E.; Martinet, P. Modeling and vision-based control of large-dimension cable-driven parallel robots using a multiple-camera setup. *Mechatronics* **2019**, *61*, 20–36. [\[CrossRef\]](#)
7. Banlue, T.; Sooraksa, P.; Noppanakeepong, S. A practical position-based visual servo design and implementation for automated fault insertion test. *Int. J. Control Autom. Syst.* **2014**, *12*, 1090–1101. [\[CrossRef\]](#)
8. Zhao, M.; Yang, Y.; Peng, X.B.; Liu, A.M.; Cheng, Y.; Wang, H.F. Neural network based visual servo for quick-change device alignment in context of fusion reactor remote maintenance. *Fusion Eng. Des.* **2022**, *181*, 113218. [\[CrossRef\]](#)
9. Zhou, S.Z.; Shen, C.; Pang, F.Y.; Chen, Z.; Gu, J.; Zhu, S.Q. Position-based visual servoing control for multi-joint hydraulic manipulator. *J. Intell. Robot. Syst.* **2022**, *105*, 33. [\[CrossRef\]](#)
10. Wang, G.J.; Qin, J.H.; Liu, Q.C.; Ma, Q.C.; Zhang, C. Image-based visual servoing of quadrotors to arbitrary flight targets. *IEEE Robot. Autom. Lett.* **2023**, *8*, 2022–2029. [\[CrossRef\]](#)
11. Asl, H.J.; Yoon, J. Robust image-based control of the quadrotor unmanned aerial vehicle. *Nonlinear Dyn.* **2016**, *85*, 2035–2048.
12. Liu, N.; Shao, X.L. Desired compensation RISE-based IBVS control of quadrotors for tracking a moving target. *Nonlinear Dyn.* **2019**, *95*, 2605–2624. [\[CrossRef\]](#)
13. Li, Y.R.; Lien, W.Y.; Huang, Z.H.; Chen, C.T. Hybrid visual servo control of a robotic manipulator for cherry tomato harvesting. *Actuators* **2023**, *12*, 253. [\[CrossRef\]](#)

14. Wang, Y.; Lang, H.X.; de Silva, C.W. A hybrid visual servo controller for robust grasping by wheeled mobile robots. *Nonlinear Dyn.* **2010**, *15*, 757–769. [\[CrossRef\]](#)
15. Gao, J.; An, X.M.; Proctor, A.; Bradley, C. Sliding mode adaptive neural network control for hybrid visual servoing of underwater vehicles. *Ocean Eng.* **2017**, *142*, 666–675. [\[CrossRef\]](#)
16. 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. [\[CrossRef\]](#)
17. Santamaria-Navarro, A.; Grosch, P.; Lippiello, V.; Sola, J.; Andrade-Cetto, L. Uncalibrated visual servo for unmanned aerial manipulation. *IEEE-ASME Trans. Mechatron.* **2017**, *22*, 1610–1621. [\[CrossRef\]](#)
18. Gong, Z.Y.; Tao, B.; Yang, H.; Yin, Z.P.; Ding, H. An uncalibrated visual servo method based on projective homography. *IEEE Trans. Autom. Sci. Eng.* **2018**, *15*, 806–817. [\[CrossRef\]](#)
19. Norouzi-Ghazbi, S.; Mehrkish, A.; Fallah, M.M.H.; Janabi-Sharifi, F. Constrained visual predictive control of tendon-driven continuum robots. *Robot. Auton. Syst.* **2021**, *145*, 103856. [\[CrossRef\]](#)
20. Qiu, Z.J.Z.; Hu, S.Q.; Liangm, X.W. Model predictive control for uncalibrated and constrained image-based visual servoing without joint velocity measurements. *IEEE Access* **2019**, *7*, 73540–73554. [\[CrossRef\]](#)
21. Huang, Y.T.; Zhu, M.; Zheng, Z.W.; Low, K.H. Homography-based visual servoing for underactuated VTOL UAVs tracking a 6-DOF moving ship. *IEEE Trans. Veh. Technol.* **2022**, *71*, 2385–2398. [\[CrossRef\]](#)
22. Zhou, Z.Y.; Zhang, R.X.; Zhu, Z.F. Robust Kalman filtering with long short-term memory for image-based visual servo control. *Multimed. Tools Appl.* **2019**, *78*, 26341–26371. [\[CrossRef\]](#)
23. Ren, X.L.; Li, H.W.; Li, Y.C. Image-based visual servoing control of robot manipulators using hybrid algorithm with feature constraints. *IEEE Access* **2020**, *8*, 223495–223508. [\[CrossRef\]](#)
24. Kang, M.; Chen, H.; Dong, J.X. Adaptive visual servoing with an uncalibrated camera using extreme learning machine and Q-learning. *Neurocomputing* **2020**, *402*, 384–394. [\[CrossRef\]](#)
25. Jin, Z.H.; Wu, J.H.; Liu, A.D.; Zhang, W.A.; Yu, L. Policy-based deep reinforcement learning for visual servoing control of mobile robots with visibility constraints. *IEEE Trans. Ind. Electron.* **2021**, *69*, 1898–1908. [\[CrossRef\]](#)
26. Aghili, F. Fault-tolerant and adaptive visual servoing for capturing moving objects. *IEEE/ASME Trans. Mechatron.* **2021**, *27*, 1773–1783. [\[CrossRef\]](#)
27. Yang, L.; Yuan, C.; Lai, G.Y. Adaptive fault-tolerant visual control of robot manipulators using an uncalibrated camera. *Nonlinear Dyn.* **2022**, *111*, 3379–3392. [\[CrossRef\]](#)
28. Fazeli, S.M.; Abedi, M.; Molaei, A.; Hassani, M.; Khosrave, M.A.; Ameri, A. Active fault-tolerant control of cable-driven parallel robots. *Nonlinear Dyn.* **2023**, *111*, 6335–6347. [\[CrossRef\]](#)
29. Nasiri, A.; Nguang, S.K.; Swain, A.; Almakhles, D. Passive actuator fault tolerant control for a class of MIMO nonlinear systems with uncertainties. *Int. J. Control* **2019**, *92*, 693–704. [\[CrossRef\]](#)
30. Wu, Y.W.; Yao, L.N. Fault diagnosis and fault tolerant control for manipulator with actuator multiplicative fault. *Int. J. Control Autom. Syst.* **2020**, *19*, 980–987. [\[CrossRef\]](#)
31. Zhao, B.; Li, Y.C. Local joint information based active fault tolerant control for reconfigurable manipulator. *Nonlinear Dyn.* **2014**, *77*, 859–876. [\[CrossRef\]](#)
32. Le, Q.D.; Kang, H.J. An active fault-tolerant control based on synchronous fast terminal sliding mode for a robot manipulator. *Actuators* **2022**, *11*, 195. [\[CrossRef\]](#)
33. Nguyen, V.C.; Tran, X.Y.; Kang, H.J. A novel high-speed third-order sliding mode observer for fault-tolerant control problem of robot manipulators. *Actuators* **2022**, *11*, 259. [\[CrossRef\]](#)
34. Zhang, J.X.; Yang, G.H. Fuzzy adaptive fault-tolerant control of unknown nonlinear systems with time-varying structure. *IEEE Trans. Fuzzy Syst.* **2019**, *27*, 1904–1916. [\[CrossRef\]](#)
35. Zhong, N.; Li, X.Z.; Yan, Z.Y.; Zhang, Z.J. A Neural Control Architecture for Joint-Drift-Free and Fault-Tolerant Redundant Robot Manipulators. *IEEE Access* **2018**, *6*, 66178–66187. [\[CrossRef\]](#)
36. Anjum, Z.; Guo, Y.; Yao, W. Fault tolerant control for robotic manipulator using fractional-order backstepping fast terminal sliding mode control. *Trans. Inst. Meas. Control* **2021**, *43*, 3244–3254. [\[CrossRef\]](#)
37. Vo, A.T.; Kang, H.J. A novel fault-tolerant control method for robot manipulators based on non-singular fast terminal sliding mode control and disturbance observer. *IEEE Access* **2020**, *8*, 109388–109400. [\[CrossRef\]](#)
38. Li, L.J.; Su, Y.X.; Kong, L.Y.; Jiang, K.; Zhou, Y.J. Finite time fault tolerant control for robot manipulators using time delay estimation and continuous nonsingular fast terminal sliding mode control. *Ocean Eng.* **2017**, *142*, 666–675.
39. Santamaria-Navarro, A.; Grosch, P.; Lippiello, V.; Sola, J.; Andrade-Cetto, L. TDE-based adaptive super-twisting multivariable fast terminal slide mode control for cable-driven manipulators with safety constraint of error. *IEEE Access* **2023**, *11*, 6656–6664.
40. Li, T.; Zhao, H. Global finite-time adaptive control for uncalibrated robot manipulator based on visual servoing. *ISA Trans.* **2017**, *68*, 402–411. [\[CrossRef\]](#) [\[PubMed\]](#)
41. Xie, C.H.; Yang, G.H. Decentralized adaptive fault-tolerant control for large-scale systems with external disturbances and actuator faults. *Automatica* **2017**, *85*, 83–90. [\[CrossRef\]](#)
42. Van, M.; Kang, H.J.; Shin, K.S. Novel quasi-continuous super-twisting high-order sliding mode controllers for output feedback tracking control of robot manipulators. *Proc. Inst. Mech. Eng. Part C-J. Mech. Eng. Sci.* **2015**, *228*, 3240–3257. [\[CrossRef\]](#)

43. Armstrong, B.; Khatib, O.; Burdick, J. The explicit dynamic model and inertial parameters of the PUMA 560 arm. *Proc. IEEE Int. Conf. Robot. Autom.* **1986**, *3*, 510–518.
44. Tang, P.; Lin, D.F.; Zheng, D.; Fan, S.P.; Ye, J.C. Observer based finite-time fault tolerant quadrotor attitude control with actuator faults. *Aerosp. Sci. Technol.* **2020**, *104*, 105968. [[CrossRef](#)]

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.