

Article

Discrete-Time Visual Servoing Control with Adaptive Image Feature Prediction Based on Manipulator Dynamics

Chenlu Liu , Chao Ye , Hongzhe Shi and Weiyang Lin * 

Research Institute of Intelligent Control and Systems, Harbin Institute of Technology, Harbin 150001, China; chenlulu.cll@gmail.com (C.L.); yechao@hit.edu.cn (C.Y.); hongzheshi.hzs@gmail.com (H.S.)

* Correspondence: wylin@hit.edu.cn

Abstract: In this paper, a practical discrete-time control method with adaptive image feature prediction for the image-based visual servoing (IBVS) scheme is presented. In the discrete-time IBVS inner-loop/outer-loop control architecture, the time delay caused by image capture and computation is noticed. Considering the dynamic characteristics of a 6-DOF manipulator velocity input system, we propose a linear dynamic model to describe the motion of a robot end effector. Furthermore, for better estimation of image features and smoothing of the robot's velocity input, we propose an adaptive image feature prediction method that employs past image feature data and real robot velocity data to adopt the prediction parameters. The experimental results on a 6-DOF robotic arm demonstrate that the proposed method can ensure system stability and accelerate system convergence.

Keywords: discrete-time system; visual servoing control; adaptive prediction; manipulator



Citation: Liu, C.; Ye, C.; Shi, H.; Lin, W. Discrete-Time Visual Servoing Control with Adaptive Image Feature Prediction Based on Manipulator Dynamics. *Sensors* **2024**, *24*, 4626. <https://doi.org/10.3390/s24144626>

Academic Editors: Daniel Galan, Ramon A. Suarez Fernandez and Francisco Javier Badesa

Received: 6 June 2024

Revised: 11 July 2024

Accepted: 15 July 2024

Published: 17 July 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

Vision perception-based robot control is of great significance in intelligent robots. In the fields of autonomous mobile robots [1–4], humanoid robots [5,6], aerial vehicles [7,8], etc., visual perception is used to improve robot autonomy, environmental adaptability, and intelligence. In addition, in the field of industrial applications, 6-DOF (six degrees of freedom) manipulators with visual servo control are widely used for tasks such as grasping, inserting, and assembling [9–11]. There are three visual servo control schemes reported in the literature, namely the (1) position-based visual servoing scheme (PBVS), (2) the image-based visual servoing scheme (IBVS), and (3) the hybrid visual servoing scheme (HVS) [12–14]. In the PBVS scheme, the features of the target object are associated with the geometric model of the object to calculate its pose. On the contrary, IBVS does not estimate the pose of the target object but directly uses image features on its controller to control robot's velocity. Therefore, it does not require precise camera calibration and geometric object models [13,15].

In classical IBVS, the points, lines, and regions of objects are often employed as image features due to their simple recognition and robustness to image noise [16,17]. Then, the controller algorithm is designed using the image Jacobian matrix to convert the image error into the desired robot motion velocity to calculate the control signal. Generally, the proportional control law is employed in classical IBVS to complete the positioning tasks [18]. Although these methods are applicable to practical visual servo tasks, they rely on the geometric features of the object, and the calculation of their Jacobian matrix requires the depth value of the target object. In addition, these methods may cause singularity problems with the image Jacobian matrix [19]. In order to apply the IBVS method to scenes where the target object is prone to deformation, many researchers have explored new image features. Image moments offer a way to describe image features without relying on the geometric characteristics of the target object. Therefore, visual servo methods based on image moments have been widely studied. F. Chaumet proposed a visual servo method

based on the general features of image moments [20]. In this study, the interaction matrices of various analytical forms based on image moments were determined. Furthermore, O. Tahri used a normalization method to process image moments and achieved decoupling control of the system by combining the scale-invariant image moments [21]. Another method called direct visual servoing, which uses features such as image histograms [22], Gaussian mixtures [23,24], and wavelets [25], has also received attention from some scholars. To avoid calculating object depth values, Liu et al. [26] constructed a Jacobian matrix that no longer requires depth information by extracting the depth variable from the interaction matrix. Ali et al. [27] proposed a depth-free IBVS method that uses the area of the power port to represent the depth information. To solve the singularity problems, many studies have been devoted to image moment-based visual servoing methods [28,29].

In addition to designing image features, extensive studies have also been conducted on visual servo controllers to achieve better control performance. Xie et al. [9] proposed a switching controller to keep image features within the view range of the camera. Li et al. [30] presented an enhanced controller based on a hybrid proportional derivative control and sliding mode control (PD-SMC) method to ensure that the robot could quickly track target objects and handle uncertainties in depth. Furthermore, considering limitations in terms of the robot's workspace and velocity, in addition to visibility constraints and parameter uncertainties, visual servoing methods based on model predictive control (MPC) have been proposed [31]. Allibert et al. [32] formulated the constraints as state constraints, as well as input and output constraints. In Hajiloo's study [33], a robust model predictive control-based IBVS controller was presented, which reduces computational costs and enables MPC to be implemented in a fast system.

Although the above methods solve the problems of classical IBVS by designing new features and different controllers, most of them are designed in continuous-time cases without considering discrete-time system performance [34]. In addition, since robots employed in IBVS are not ideal rigid systems, their dynamic models should also be considered. This will benefit us for predicting image features in discrete-time visual servoing systems. Conticelli et al. derived an interaction model between a camera and an object in discrete time and implemented positioning tasks by designing a visual controller in discrete time [35]. Bjerkeng et al. [36] studied the stability of robot control based on sensor data feedback in discrete time. The authors derived the stability results of visual servoing considering the input/output time delays and the robot dynamics. The result of small-gain proportional control was obtained for the high-delay problem of sensor feedback. He [37] designed an input-mapping method to achieve stability in a discrete visual servo system. Past image errors and camera motion data were employed to calculate the uncertain part of the model. Although these methods have developed visual servo systems from a discrete perspective and designed controllers to ensure system stability, they have not improved system performance by increasing the frequency of image information feedback.

Unlike continuous visual servo methods, this paper describes a visual servo system in discrete time. Furthermore, by analyzing the inner–outer loop control architecture of a visual servo system, the influence of sampling frequency is presented. In order to improve the performance of discrete visual servo systems, this paper develops a discrete-time visual servo controller with adaptive image feature prediction. In the proposed method, we consider both image processing delay and robot dynamics in the system. Past image feature data and robot end-effector velocity data are used to adaptively adjust predictor parameters. The main contributions of this study are summarized as follows:

- (1) We present a visual servoing method from the perspective of discrete-time analysis. Based on the visual servo inner–outer loop control architecture, the impact of low-frequency visual feedback on system performance is analyzed.
- (2) A linear dynamic model is investigated to describe the movement of the robot under the velocity control mode. By using this model, the actual velocity of the robot end effector can be more accurately estimated.

- (3) An adaptive image feature prediction method is proposed to predict the positions of image features during the image sampling and processing time. Past image feature data and real robot velocity data are employed to adjust the predictor parameters.

The rest of this paper is organized as follows. Section 2 describes the discrete-time visual servo model and the problem in a practical discrete-time visual servoing system. Section 3 presents the adaptive image feature prediction based on the linear dynamics model. Section 4 reports the experimental results. Finally, the conclusion is presented in Section 5.

2. System Description and Control Architecture

2.1. System Description

Consider fixing the camera on the end effector of a collaborative robot, as shown in Figure 1. In positioning tasks, the purpose of the IBVS method is to control the robot's motion until the image features of the object move to the desired feature position in the image, which is typically defined as

$$\mathbf{e}(t) = \mathbf{s}(t) - \mathbf{s}^*, \quad (1)$$

where $\mathbf{s}(t)$ is the image feature vector captured by vision sensors at time t , and vector $\mathbf{s}^*$ is the desired value of the features. $\mathbf{e} \in \epsilon$ is the vector of image error variables, with ϵ being a domain of $\mathbb{R}^m$. As shown above, in the IBVS scheme, the task error ($\mathbf{e}$) is defined in image space, and the input of the manipulator controller is defined by joint coordinates or Cartesian coordinates. Let $\bar{r}$ be the location of the target object where the features of the object (points, lines, etc.) naturally exist. According to [13], selecting appropriate object features and applying the projection mapping principles, we can obtain

$$\mathbf{s} = F(\bar{r}), \quad (2)$$

where F is the perspective projection mapping. Take the derivative of Equation (2) and then the velocities of image features are described as

$$\dot{\mathbf{s}} = \frac{\partial \mathbf{s}}{\partial \bar{r}} \frac{d\bar{r}}{dt} = \mathbf{L}\mathbf{V}_c, \quad (3)$$

where $\dot{\mathbf{s}}$ represents the image feature velocities in the image plane and $\mathbf{V}_c = [v_x, v_y, v_z, w_x, w_y, w_z]^\top$ is the camera velocity screw with respect to the target object. $\mathbf{L} \in \mathbb{R}^{k \times 6}$ is generally called the image Jacobian or the interaction matrix [4,13,18]. If the system uses image points as features, $\mathbf{L}$ can be described as

$$\mathbf{L} = \begin{bmatrix} -\frac{\lambda}{z} & 0 & \frac{u}{z} & \frac{uv}{\lambda} & -\frac{\lambda^2 + u^2}{\lambda} & v \\ 0 & -\frac{\lambda}{z} & \frac{v}{z} & \frac{\lambda^2 + v^2}{\lambda} & -\frac{uv}{\lambda} & -u \end{bmatrix}, \quad (4)$$

where z is the height of image features along the optical axis in the camera coordinates, and λ is the focal length of the camera lens. u and v are the positions of image features in the image space [13].

In a discrete-time system, the image feature position at time k is given by $\mathbf{s}(k)$. Let us set the camera sample time as T_i , and the continuous time is given by $t = kT_i$. For simplicity, the following formula uses k instead of kT_i to represent the current time. To obtain the discrete-time model, Equation (1) is expressed as

$$\mathbf{e}(k) = \mathbf{s}(k) - \mathbf{s}^* \quad (5)$$

Furthermore, with Equations (3) and (5), the discrete-time model can be obtained as

$$\mathbf{e}(k+1) = \mathbf{e}(k) + \mathbf{T}\mathbf{L}(k)\mathbf{V}_c(k) \quad (6)$$

To converge image errors to zero, a proportional control scheme is often adopted in classical visual servoing controllers [18], in which

$$\dot{\mathbf{e}}(k) = -\mathbf{K}_p \mathbf{e}(k) \quad (7)$$

Combining Equations (6) and (7), the camera velocity can be expressed as

$$\mathbf{V}_c(k) = -\mathbf{K}_p \mathbf{L}^+(k) \mathbf{e}(k), \quad (8)$$

in which $\mathbf{L}^+$ is the pseudo-inverse of $\mathbf{L}$.

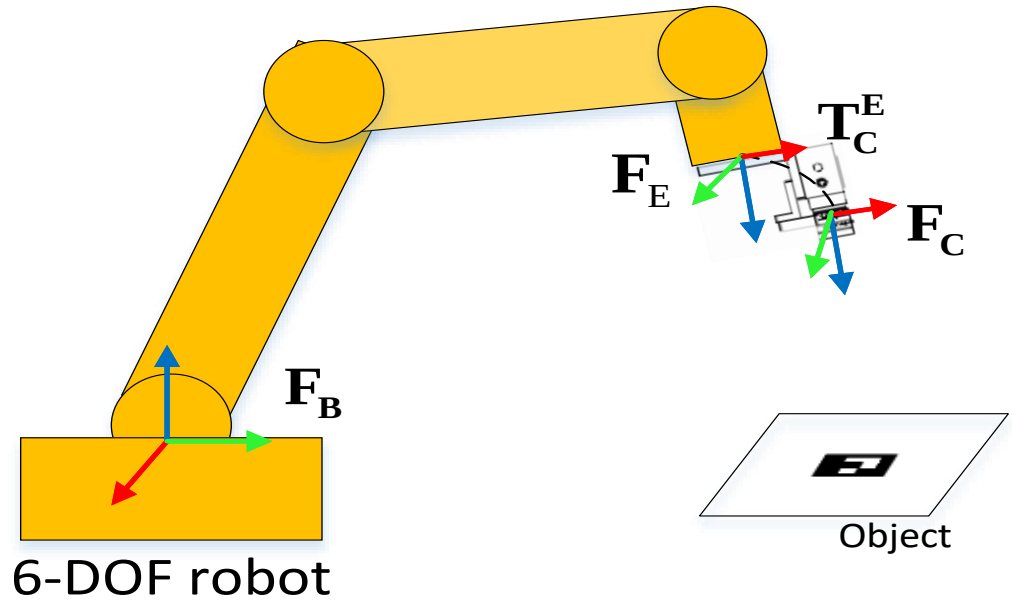


Figure 1. Coordinate system definition in the visual servoing system.

2.2. Control Architecture

Defining $\mathbf{q} \in \mathbb{R}^n$ as the robot configuration variables, the IBVS scheme can be expressed as follows:

$$\mathbf{f} : \mathbf{q} \subseteq \mathbb{R}^n \mapsto \mathbf{e} \subseteq \mathbb{R}^m, \mathbf{e} = \mathbf{f}(\mathbf{q}) \quad (9)$$

For example, in an IBVS system based on a manipulator, $\mathbf{e}$ is the vector of image feature errors, and $\mathbf{q}$ is the vector of robot joint position or robot end-effector position.

In IBVS robotic systems, the classic control frame is an inner-loop/outer-loop architecture, as shown in Figure 2. The inner loop is typically composed of a robot controller and robot motor servo drivers. The controller controls the robot's motion by sending position, velocity, or torque commands to the robot drive. Based on the required speed command, the robot controller uses a robot kinematics or dynamics model to calculate the joint velocities and sends these to the robot. Due to low computational complexity and short transmission delay, the control frequency of the inner loop is usually as high as 500–4000 Hz.

The outer loop, based on image information feedback, executes task layer applications such as grasping, placing, and object tracking. Considering the time required for image data acquisition and processing, the outer loop typically has high computational costs and a low control frequency, typically in the range of 30 Hz to 200 Hz.

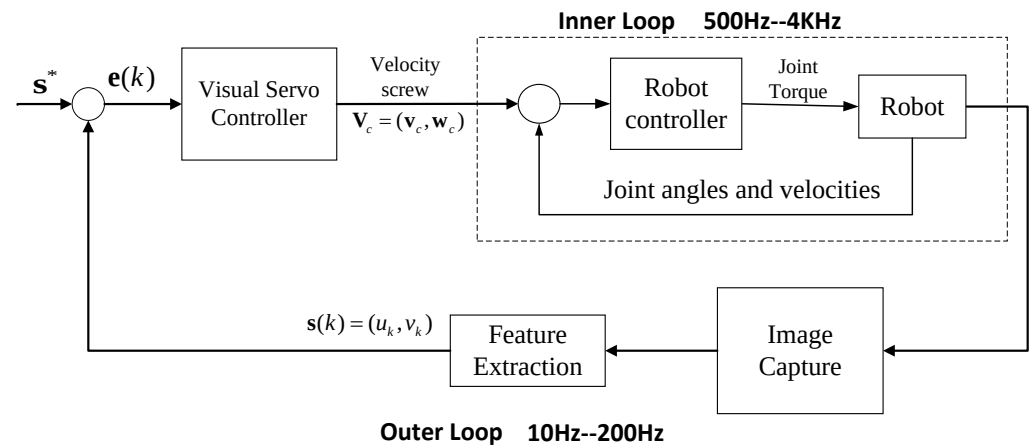


Figure 2. The classical visual servo control architecture.

As shown in Figure 3, since the robot control frequency exceeds the image sampling rate, the robot uses the image features of the previous moment during each image feature sampling period until it obtains a new image. T_O is the outer-loop cycle time, and T_I is the inner-loop cycle time. Therefore, Equation (8) can be rephrased as

$$\mathbf{V}_c(1 + (k - 1)N) = -\mathbf{K}_p \mathbf{L}^+ \mathbf{e}(k), \quad (10)$$

where $N = T_O/T_I$ and

$$\mathbf{V}_c(1 + (k - 1)N) = \mathbf{V}_c(m + (k - 1)N), \quad (11)$$

in which $m = 1, 2, \dots, N - 1$, and $k = 1, 2, \dots$.

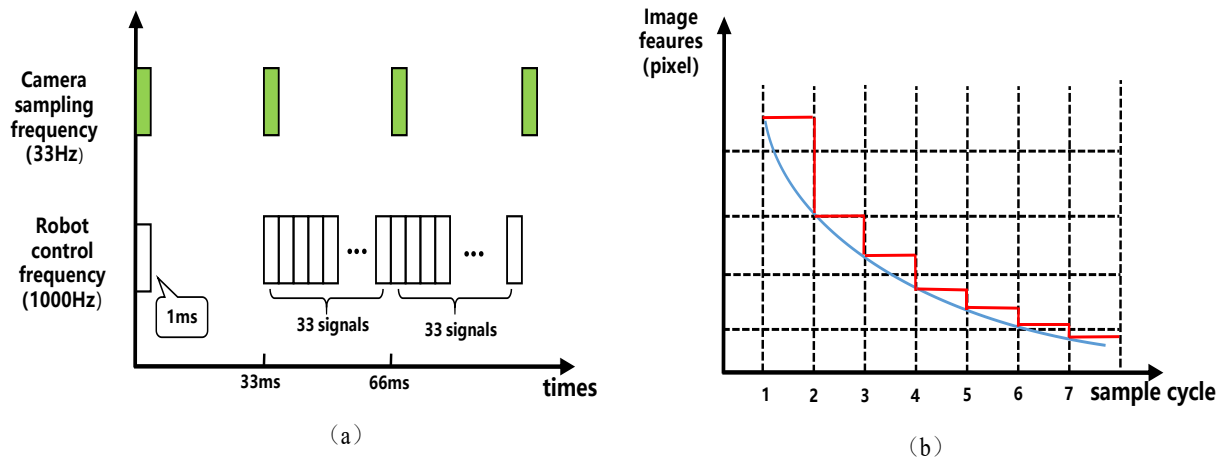


Figure 3. System signals and image feature values. (a) Camera sampling and robot control timing diagram. (b) Feedback values of image features and actual image features values. Red line: image feature feedback value. Blue line: actual image feature value.

It is noted that the time delay has a great influence on the visual servoing system. The controller uses the same image feature signal within N control cycles, which may lead to system stability and oscillation issues in the worst-case scenarios. Although many scholars have studied image error estimation, they have not explicitly considered the difference between the control period and the image feedback period.

3. Adaptive Image Feature Prediction Based on Robot Velocity Linear Dynamic Model

In this section, we introduce the linear dynamic model of the robot end-effector velocity, which benefits us in predicting the image features. Then, the implementation of adaptive image feature prediction is introduced in detail.

3.1. Linear Dynamic Model of Robot and Camera Motion

Let frame $\mathbf{F}_E$ link to the robot end effector and $\mathbf{F}_C$ link to the camera. The displacement between the robot end-effector coordinate system ($\mathbf{F}_E$) and the camera coordinate system ($\mathbf{F}_C$) can be denoted as $\mathbf{P}_C^E = (x_c, y_c, z_c) \in \mathbb{R}^3$, employing $\mathbf{R}_C^E \in \mathbb{R}^{3 \times 3}$ as the rotation matrix from $\mathbf{F}_C$ to $\mathbf{F}_E$. The feature point in the camera coordinate system is denoted as $\mathbf{P}_C$, and it is represented as $\mathbf{P}_E$ in the robot end-effector coordinate system, so we have

$$\mathbf{P}_E = \mathbf{T}_C^E * \mathbf{P}_C, \quad (12)$$

where $\mathbf{T}_C^E$ is the homogeneous transformation matrix, and

$$\mathbf{T}_C^E = \begin{bmatrix} \mathbf{R}_C^E & \mathbf{P}_C^E \\ 0 & 1 \end{bmatrix} \quad (13)$$

$\mathbf{V}_e$ is defined as the robot's end-effector velocities. Considering the camera is fixedly connected to the robot end effector, the robot end-effector velocities are

$$\mathbf{V}_e = \mathbf{H}\mathbf{V}_c, \quad (14)$$

where

$$\mathbf{H} = \begin{bmatrix} \mathbf{R}_C^E & S(\mathbf{P}_C^E)\mathbf{R}_C^E \\ 0_{3 \times 3} & \mathbf{R}_C^E \end{bmatrix}, \quad (15)$$

where S is the skew symmetric matrix and $S(\mathbf{P}_C^E)$ is given by

$$S(\mathbf{P}_C^E) = \begin{bmatrix} 0 & -z_c & y_c \\ z_c & 0 & -x_c \\ -y_c & x_c & 0 \end{bmatrix} \quad (16)$$

Ideally, the output of the robot's velocity mode is expected to be a step response, which means that

$$\bar{\mathbf{V}}_e(n) = \mathbf{V}_e^{ref}(n), \quad (17)$$

where $\mathbf{V}_e^{ref}(n)$ is the expected velocity output by the visual servo controller, and $\bar{\mathbf{V}}_e(n)$ is the real velocity of the robot end effector.

In practice, the body of a robot is flexible, and its velocity control mode is not ideal. Considering the nonlinear part of its response, we propose a linear model taking into account robot dynamics that is suitable for robots that compute torque in the robot controller [36,38].

Assuming the end velocity of the robot is a uniform acceleration model, within the robot control sample time (T_r), the robot's motion distance (L) is

$$L = \frac{\mathbf{V}_e(n-1) + \mathbf{V}_e(n)}{2} t_1 + \mathbf{V}_e(n) t_2, \quad (18)$$

where t_1 is the acceleration time, t_2 is the time for uniform motion, and $T_r = t_1 + t_2$. $\mathbf{V}_e(n)$ is the actual velocity of the robot. With knowledge of the definition of velocity in physics, the real velocity can be obtained as follows:

$$\begin{aligned} \bar{\mathbf{V}}_e(n) &= L/T \\ &= \frac{t_1}{2T} \mathbf{V}_e(n-1) + \frac{t_1 + 2t_2}{2T} \mathbf{V}_e(n) \end{aligned} \quad (19)$$

Let $\alpha \in (-1, 1)$ be the convergence rate, employing β as the gain value for the expected velocity. According to Equation (19), the linear dynamic model of the robot velocity is given by

$$\bar{\mathbf{V}}_e(n) = \alpha \mathbf{V}_e(n-1) + \beta \mathbf{V}_e^{ref}(n) \quad (20)$$

When $\alpha = 0$, the robot velocity controller is perfect and can converge within one step. If the value of α approaches 1, the system is overdamped. For a high-performance robot in velocity control mode, $\beta \approx 1$, indicating that the robot can achieve the required velocity command with a small error. In practice, the robot motion control system is stable. The deviation between the desired velocity and the actual motion velocity is bounded. Thus, we have

$$\|\bar{\mathbf{V}}_e(n)\| \leq \eta \|\mathbf{V}_e^{ref}(n)\|, \quad (21)$$

where $\eta \in (0, \eta_1)$, $\eta_1 \in \mathbb{R}$.

Currently, most robots provide velocity operation mode in Cartesian space, which greatly facilitates the operation of the robot. Therefore, we can directly send velocity commands to the robot without considering complex joint velocity calculations. According to Equation (14), we have

$$\bar{\mathbf{V}}_c(n) = \alpha \mathbf{H}^+ \mathbf{V}_e(n-1) + \beta \mathbf{H}^+ \mathbf{V}_e^{ref}(n) \quad (22)$$

3.2. Adaptive Image Feature Prediction

In this subsection, we assume that the robot does not have singularity problems. Since the majority of time is spent on image capture and processing in the outer loop, we set $\mathbf{T}_O = \mathbf{T}_i$ and $\mathbf{T}_I = \mathbf{T}_r$. In most IBVS systems, the time ($\mathbf{T}_i$) is much longer than the robot controller time ($\mathbf{T}_r$). Therefore, we use the robot velocity information at $n\mathbf{T}_r$ to estimate the image error. Considering the model dynamics, according to Equations (14) and (20), we have

$$\mathbf{e}(n+1) = \mathbf{e}(n) + \mathbf{T}_L(n) \mathbf{H}^+ \bar{\mathbf{V}}_e(n) \quad (23)$$

When $n = k(\mathbf{T}_i/\mathbf{T}_r)$, $n, k = 1, 2, \dots$, we can obtain the image error directly from the camera. If $n \neq k(\mathbf{T}_i/\mathbf{T}_b)$, we employ Equation (23) to estimate the image error.

It is important to accurately update α and β to estimate image feature positions. It should also be noted that due to the uncertainty of robot dynamics, α and β vary with robot posture.

Here, an effective method based on the least squares method is proposed for adaptive adjustment of α and β . The image feature errors and robot end-effector velocities are employed as observational data. In the first control cycle of the robot control inner loop, according to Equation (23), we can obtain

$$\mathbf{e}(1) - \mathbf{e}(0) = \mathbf{T}_r \mathbf{L}(1) \mathbf{H}^+ (\alpha \mathbf{V}_e(0) + \beta \mathbf{V}_e^{ref}(1)) \quad (24)$$

Similarly, the image feature errors are calculated sequentially in other control cycles as

$$\begin{aligned} \mathbf{e}(2) - \mathbf{e}(1) &= \mathbf{T}_r \mathbf{L}(2) \mathbf{H}^+ (\alpha \mathbf{V}_e(1) + \beta \mathbf{V}_e^{ref}(2)) \\ \mathbf{e}(3) - \mathbf{e}(2) &= \mathbf{T}_r \mathbf{L}(3) \mathbf{H}^+ (\alpha \mathbf{V}_e(2) + \beta \mathbf{V}_e^{ref}(3)) \\ &\vdots \\ \mathbf{e}(n-m) - \mathbf{e}(n-m-1) &= \mathbf{T}_r \mathbf{L}(n-m) \mathbf{H}^+ (\alpha \mathbf{V}_e(n-m-1) + \beta \mathbf{V}_e^{ref}(n-m)) \end{aligned} \quad (25)$$

Let $\mathbf{F}(n) = \mathbf{T}_r \mathbf{L}(n) \mathbf{H}^+$; for the above iterative process, the prediction equation is

$$\mathbf{e}(n) - \mathbf{e}(n-1) = \mathbf{F}(n) (\alpha \mathbf{V}_e(n-1) + \beta \mathbf{V}_e^{ref}(n)). \quad (26)$$

In order to better represent the parameters of the linear dynamic model of robot end velocity, we set $Z(n) = \mathbf{F}^{-1}(n)[\mathbf{e}(n) - \mathbf{e}(n-1)]$ as the system output, and Equation (26) can be rewritten as

$$Z(n) = [\mathbf{V}_e(n-1), \mathbf{V}_e^{ref}(n)] \begin{bmatrix} \alpha \\ \beta \end{bmatrix} \quad (27)$$

Assuming that the length of the moving window is K and $\hat{\boldsymbol{\theta}} = [\alpha, \beta]^T$, we collect Equations (25) and (26) in a compact matrix form as

$$\mathbf{Z}(n) = \bar{\mathbf{F}}^{-1} \mathbf{Y} = \Phi \hat{\boldsymbol{\theta}}, \quad (28)$$

where

$$\mathbf{Z}(n) = [Z(n), Z(n-1), \dots, Z(n-K+1)]^T \quad (29)$$

$$\bar{\mathbf{F}} = \begin{bmatrix} \mathbf{F}(n) & 0 & \dots & 0 \\ 0 & \mathbf{F}(n-1) & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \mathbf{F}(n-K+1) \end{bmatrix}_{K \times K} \quad (30)$$

$$\mathbf{Y} = \begin{bmatrix} e(n+1) - e(n) \\ e(n) - e(n-1) \\ \vdots \\ e(n-K+2) - e(n-K+1) \end{bmatrix}_{K \times 1} \quad (31)$$

$$\Phi = \begin{bmatrix} \mathbf{V}_e(n) & \mathbf{V}_e^{ref}(n+1) \\ \mathbf{V}_e(n-1) & \mathbf{V}_e^{ref}(n) \\ \vdots & \vdots \\ \mathbf{V}_e(n-K+1) & \mathbf{V}_e^{ref}(n-K+2) \end{bmatrix}_{K \times 2} \quad (32)$$

We employ $\hat{Z}(n)$ as the measurement value of the system. When $n = k(T_i/T_r)$, $\hat{Z}(n)$ is

$$\hat{Z}(n) = \mathbf{F}^{-1}(n)[\hat{\mathbf{e}}(n) - \mathbf{e}(n-1)] \quad (33)$$

where $\hat{\mathbf{e}}(n)$ is the actual image error at the image sample time. During the non-sampling period of the camera, let $\hat{Z}(n) = Z(n)$. Thus, the measurement value ($\hat{Z}(n)$) can be expressed as

$$\hat{Z}(n) = \begin{cases} \mathbf{F}^{-1}(n)[\hat{\mathbf{e}}(n) - \mathbf{e}(n-1)] & n = k(T_i/T_r) \\ Z(n) & n \neq k(T_i/T_r) \end{cases} \quad (34)$$

We set $\hat{\mathbf{Z}}$ as the vector representation of $\hat{Z}(n)$ in K control cycles. We define the cost function as

$$\mathbf{J} = \sum_{n=1}^K [\hat{Z}(n) - Z(n)]^2 \quad (35)$$

By using the least squares method, the $\hat{\boldsymbol{\theta}}$ parameter can be obtained as

$$\hat{\boldsymbol{\theta}} = (\Phi^T \Phi)^{-1} \Phi^T \hat{\mathbf{Z}} \quad (36)$$

Figure 4 shows our overall visual servo system. In contrast to basic visual servoing methods, the proposed feature prediction method combines past image data and robot end-effector velocity to predict the current image features. The method steps are given in Algorithm 1.

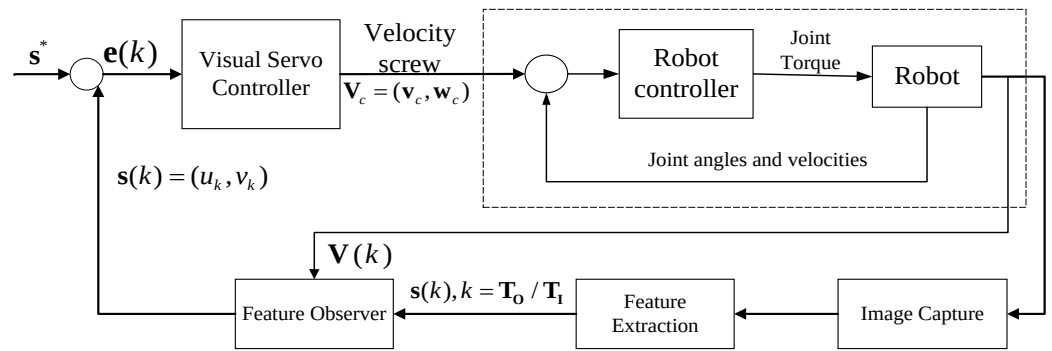


Figure 4. Discrete-time visual servo control system with adaptive image feature prediction.

Algorithm 1 Visual Servoing Controller With Adaptive Image Feature Prediction

Input: the robot end-effector velocity $\mathbf{V}_e(n)$ and the image features $\mathbf{e}(k)$

Output: the robot end-effector velocity command $\mathbf{V}_e^{ref}(n)$;

- 1: Set initial state, such as $\alpha = 0, \beta = 1$, the window length K ($K > (T_i/T_r)$), the control gain $\mathbf{K}_p$, the final image feature errors δ , the robot control time $n = 1$ and the image feature sampling time $k = 1$. $\mathbf{H}$ is the constant matrix and $\mathbf{e}(1)$ can be directly obtained from the camera.
 - 2: Calculate the robot end-effector command $\mathbf{V}_e^{ref}$ by (8) and (14).
 - 3: If $kT_i = nT_r$, the image error $\mathbf{e}$ is obtained through the camera and $k \leftarrow k + 1$. Otherwise, the image error $\mathbf{e}$ is predicted through (20) and (23).
 - 4: Update the matrix $\mathbf{Y}$ and Φ by the past data. If $n > K$, then calculate the $\hat{\theta}$ by (36).
 - 5: If $\|\mathbf{e}\| > \delta$, update $n \leftarrow n + 1$ and come back to Setp. 2.
-

3.3. Stability Analysis

The Lyapunov function is designed as

$$\mathbf{V}(n) = \frac{1}{2} \|\mathbf{e}(n)\|^2 = \frac{1}{2} \mathbf{e}^T(n) \mathbf{e}(n) \quad (37)$$

Furthermore, we define $\Delta \mathbf{V}(n) = \mathbf{V}(n+1) - \mathbf{V}(n)$. According to Equations (8), (21), and (23), we have

$$\Delta \mathbf{V}(n) = \frac{1}{2} (\|(\mathbf{I} - \eta \mathbf{T}_r \mathbf{K}_p) \mathbf{e}(n)\|^2 - \|\mathbf{e}(n)\|^2) \quad (38)$$

To ensure system stability, the following condition should be satisfied [37]:

$$\|\mathbf{I} - \eta \mathbf{T}_r \mathbf{K}_p\|^2 < \mathbf{I} \quad (39)$$

In practice, η is close to 1, and $\mathbf{T}$ is very small. We can choose the appropriate $\mathbf{K}_p$ to ensure the system's stability and improve its performance.

4. Experiments

Several comparative experiments were designed to demonstrate the impact of the proposed method on system performance. We used a lightweight collaborative robot and a camera fixed on a 6-DOF collaborative robot end effector. When the robot is in velocity mode, the controller can send speed commands to control the movement of the robot's end effector. We also selected the corner points of the ArUco marker as the feature points. The experimental equipment is shown in Figure 5.



Figure 5. Experimental setup.

The robot has a velocity control mode with a communication frequency of 1000 Hz. A Galaxy camera is employed to capture the position of the feature points, with a resolution of 1920×1080 . Considering the image acquisition time and image processing time, the feedback frequency of image features is about 30 Hz. According to the design and calibration parameters, $\mathbf{T}_C^E$ is

$$\mathbf{T}_C^E = \begin{bmatrix} 0 & 1 & 0 & 0.05 \\ -1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0.08 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (40)$$

We conducted comparative experiments of visual servoing using the following three methods:

- (1) Method 1 utilized the classical visual servoing scheme (VSC) [18]. In VSC, the four points are employed as image features. Due to the easy recognition of image features, the simple interaction matrix of point features, and high computational efficiency, this scheme has been widely applied in practice.
- (2) Method 2 (VSC-OB) employed the image prediction method based on the observer, which only uses Equation (6) and does not consider the robot dynamic model.
- (3) Method 3 (VSC-P) utilized the proposed adaptive image prediction based on the input-mapping method in combination with the robot's dynamic model.

The robot moves from the initial position where the joint angles are ($\mathbf{q} = [-100.5, 29.6, 97.1, -36.68, -90.37, -10.26]$) and the image points are $((1214, 581), (1214, 960), (836, 960))$ and $(836, 581)$. The length of the moving window is set as $K = T_i/T_r = 33$. We only considered the translational motion of the robot in the experiments, disregarding the rotational movements.

4.1. Controller with Smaller Proportional Gain

In this subsection, a comparative experiment with the controller selecting a small proportional gain is designed to compare the three different methods. In this comparative experiment, the gain is set as 15.

Figure 6 shows the feature trajectory within the image space of the three methods. It can be observed that all the methods are capable of accomplishing the visual servoing task of moving the robot from the start position to the desired location. At a small proportion of gain, the trajectories in the image space are smooth and close to straight lines.

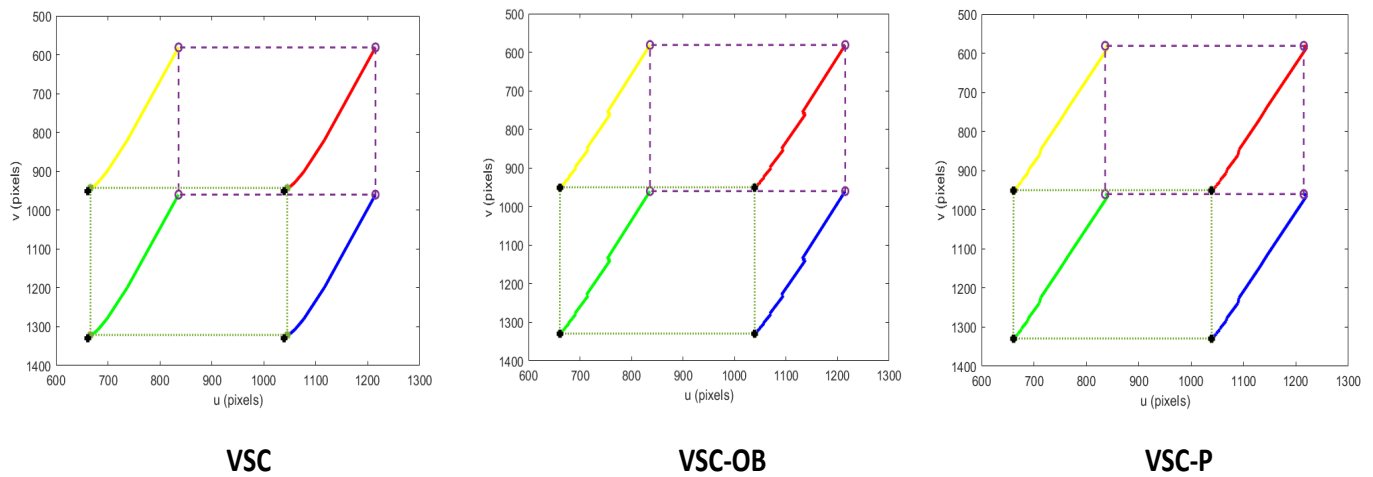


Figure 6. Image trajectory from the start position (○) to the desired position (+) in image space using a controller with a small proportional gain.

Figure 7 displays the image feature error, and Figure 8 shows the end-effector velocities and joint velocities of the robot. We can see that with the application of a small proportional gain controller, all three visual servo methods can cause the robot's end-effector velocity to converge. However, due to the low frequency of image information feedback, the VSC scheme exhibits a stepped error in its image features and experiences significant speed fluctuations during the servo process. On the contrary, owing to the presence of the image feature observer, VSC-OB can predict image features and minimize the velocity changes of the robot end effector. VSC-P integrates an image feature observer and considers the robot's dynamic characteristics, resulting in smoother velocity control at the robot end effector.

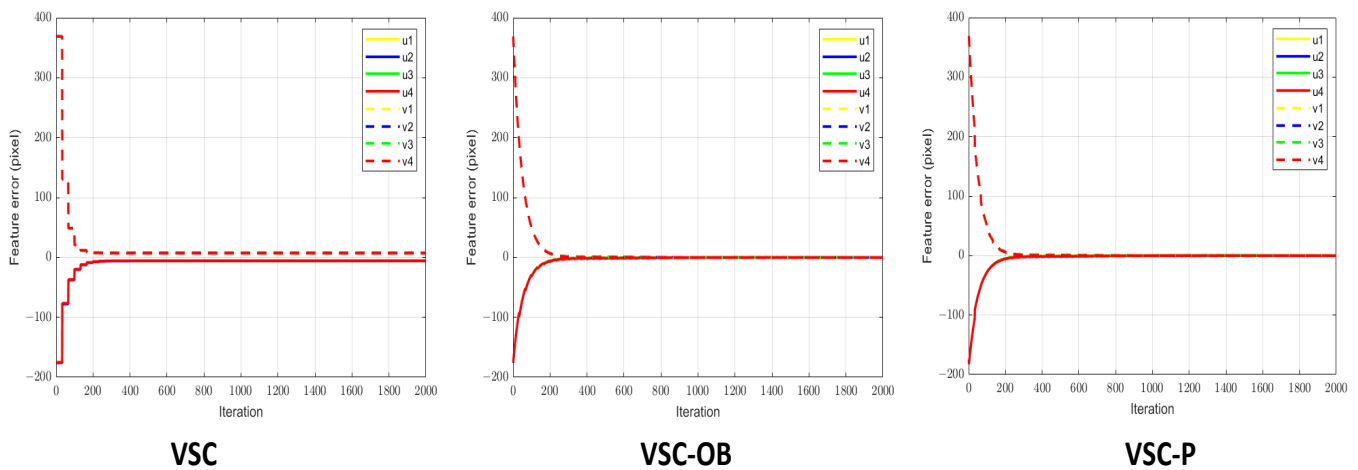
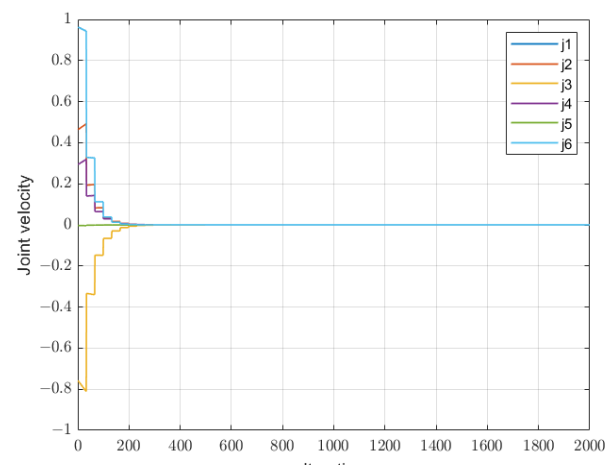
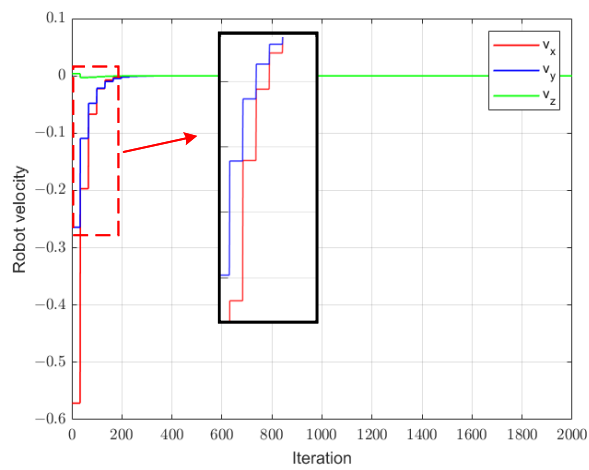
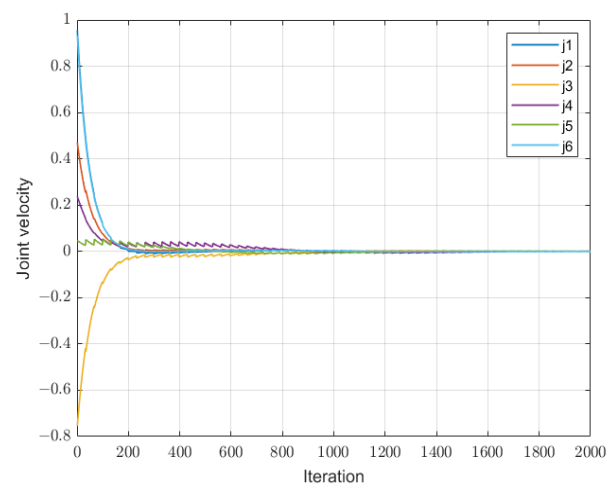
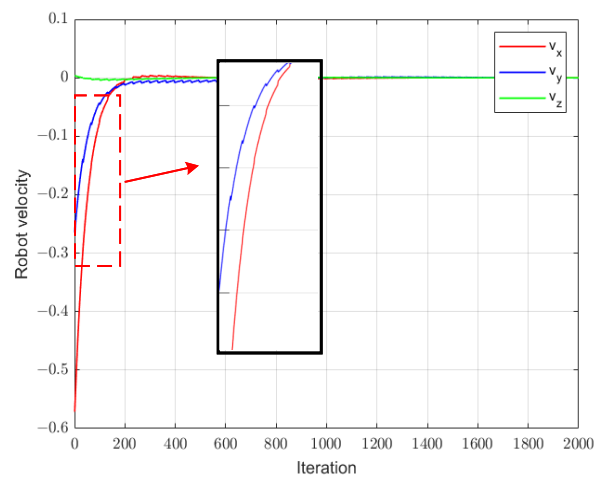


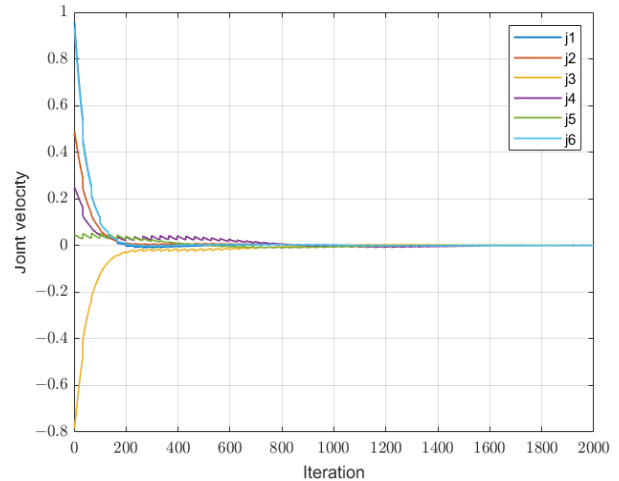
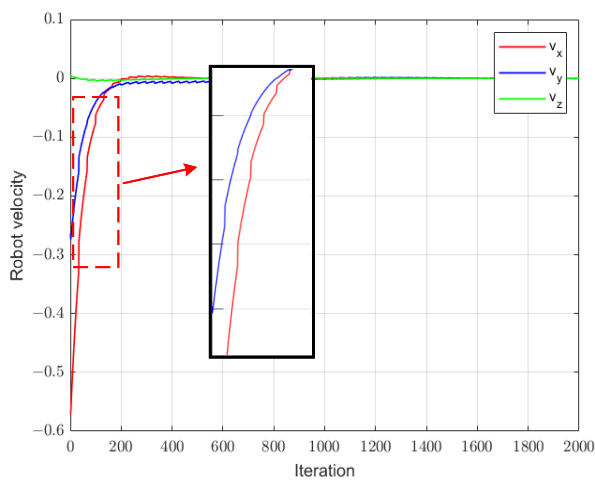
Figure 7. Image feature errors with a small proportional gain controller.



(a)



(b)



(c)

Figure 8. Robot end-effector and joint velocities with a small proportional gain controller. (a) VSC method. (b) VSC-OB method. (c) VSC-P method.

4.2. Controller with Large Proportional Gain

To explore the control performance of the proposed method with larger proportional gains, a controller with a high proportional gain value is used in this subsection. In this experiment, the proportional gain value is set as 50.

Figure 9 illustrates the moving trajectory of the robot end effector using three different schemes, while Figure 10 shows the feature trajectory in the image space. It can be observed that with a larger proportional gain controller, the VSC method induces vibration as the robot end effector approaches the target position. In the VSC-OB method, the trajectory of the robot end effector experiences a slight jitter. Compared to the first two methods, the proposed method achieves a smoother trajectory for the robot's motion and ensures that there are no oscillations when the robot approaches the target position.

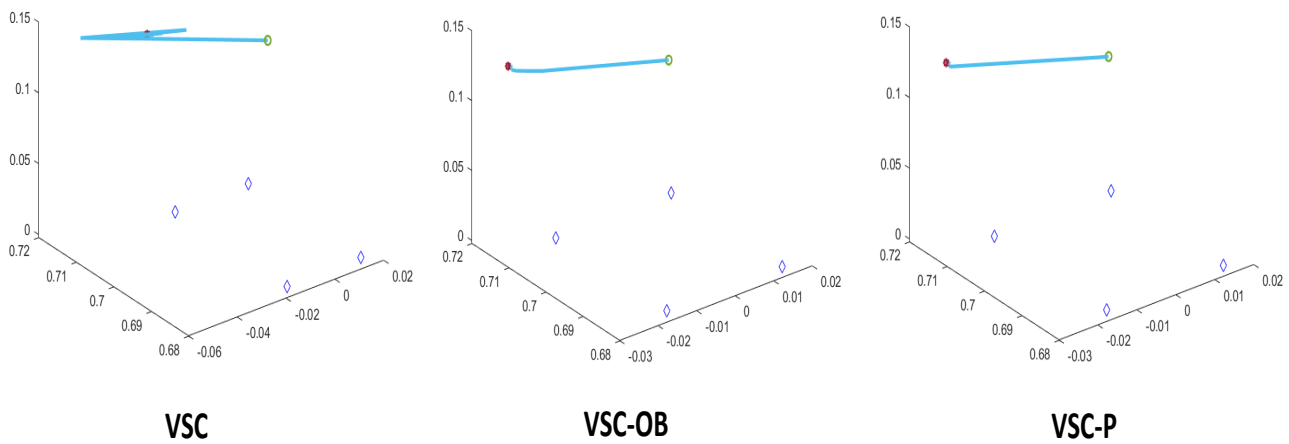


Figure 9. Robot end-effector moving trajectory from the start position (o) to the desired position (*).

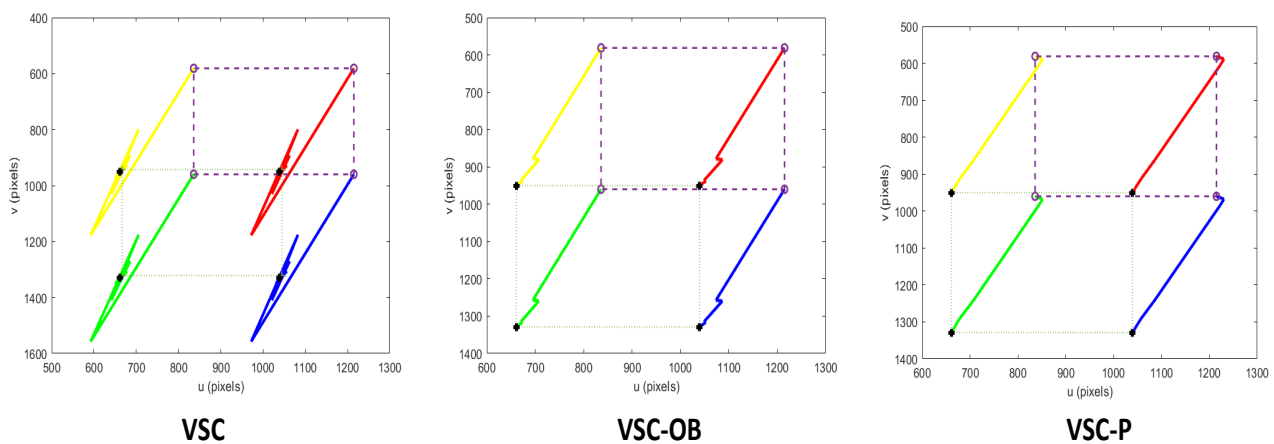


Figure 10. Image trajectory from the start position (o) to desired position (+) in image space with a large proportional gain controller.

The image errors of the three methods are shown in Figure 11, and the robot end-effector and joint velocities are shown in Figure 12a–c. It is obvious that in the case of large gain values, the VSC scheme causes oscillation of the robot's end-effector movement during the visual servo process. In contrast, because of the precise prediction of changes in image error during the control cycle, the control signal can decrease as the image error diminishes. Therefore, both the VSC-OB method and the proposed VSC-P method can not only achieve convergence but also improve the convergence speed of the system when employing a larger proportion of controllers. Furthermore, we can see that compared with the VSC-OB method, the proposed method takes into account the dynamic characteristics

of the robot's velocity mode, making image error prediction more accurate. Therefore, the robot's end-effector motion is smoother than under the VSC-OB method.

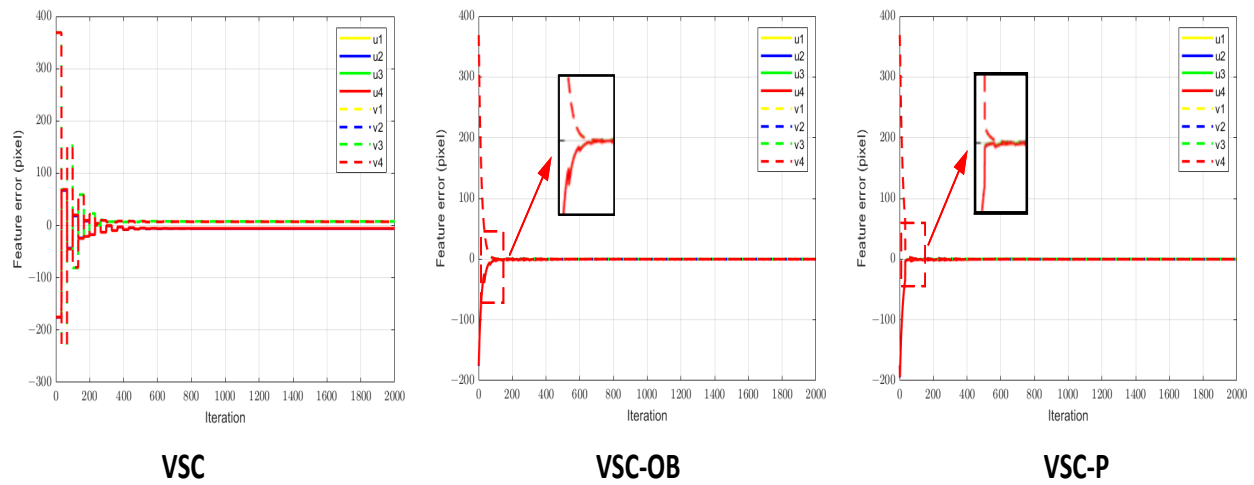


Figure 11. Image feature errors with a large proportional gain controller.

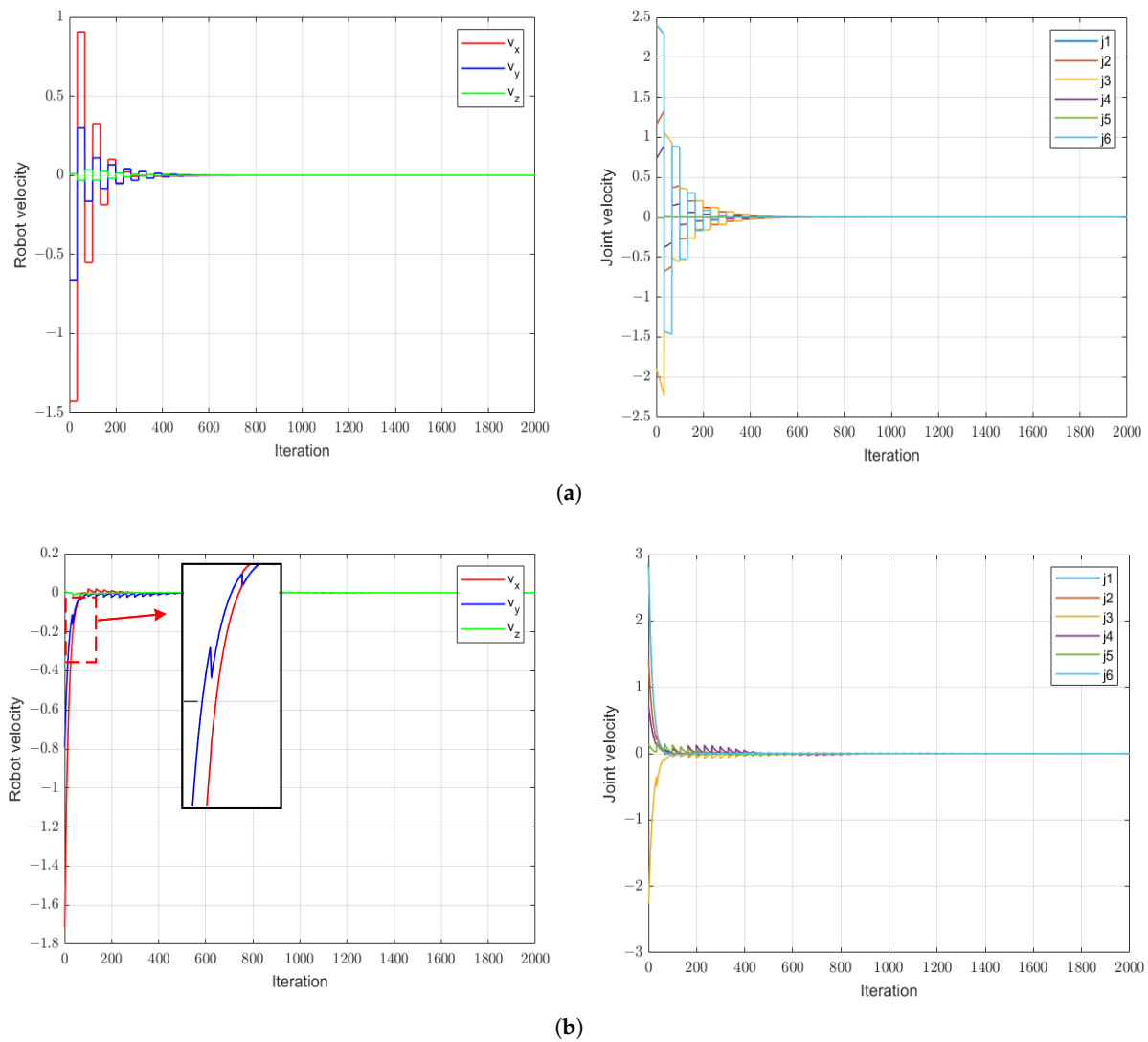


Figure 12. Cont.

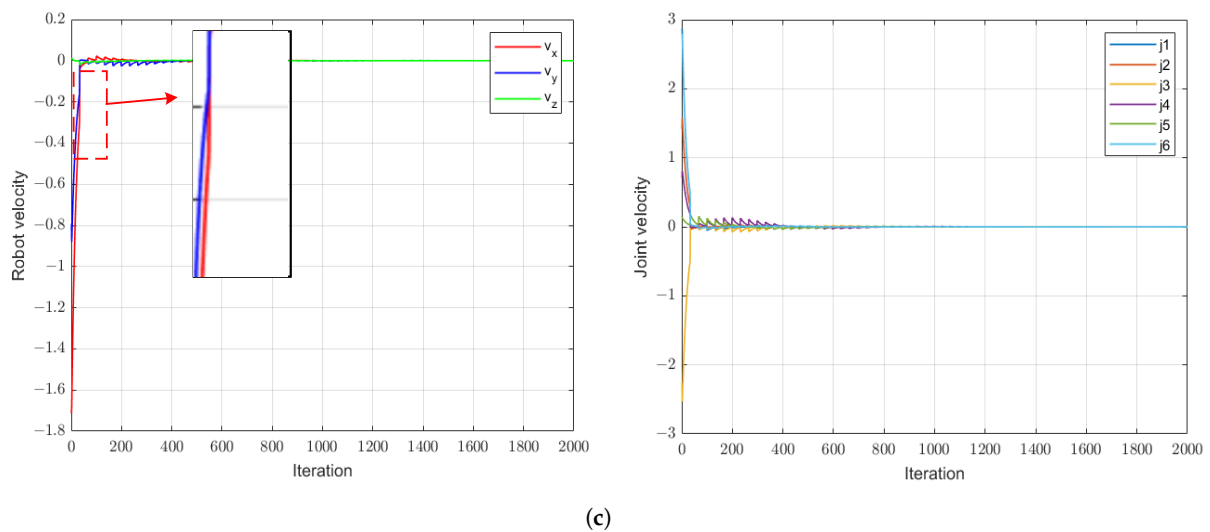


Figure 12. Robot end-effector and joint velocities with a large proportional gain controller. (a) VSC method. (b) VSC-OB method. (c) VSC-P method.

5. Conclusions

In this paper, a visual servoing control method is investigated from the perspective of a discrete-time model. To solve the time delay issue of the outer loop, image features are directly estimated in the image plane, which is suitable for scenes with low camera sampling frequency and complex image processing. This work designed an adaptive image feature prediction method by using past image feature data and robot end-effector velocity data. The controller with adaptive feature prediction smooths the velocity input of the IBVS system and still maintains the stability of the system under large gains. Compared with the classical IBVS scheme, the VSC-P method is more applicable for situations with a large outer-loop time delay. For scenarios that require high resolution and high-precision positioning, such as industrial assembly, our method can greatly reduce the problem of slow system convergence caused by long image computation time. It should be noted that the proposed method does not take into account cases of image feature feedback errors or image information loss. Therefore, in future work, we will further investigate the estimation of image features at the sampling time to reduce the impact of image feature information loss or image noise on system performance.

Author Contributions: Conceptualization, W.L.; Methodology, C.L.; Software, C.L. and C.Y.; validation, H.S.; Writing—Original Draft Preparation, C.L.; Writing—Review and Editing, C.L. and C.Y.; Visualization, H.S.; Supervision, W.L.; Funding Acquisition, W.L. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the key R&D projects in Sichuan province (No. 2023YFG0061).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Data are contained within the article.

Acknowledgments: The authors are thankful to the anonymous reviewers whose comments helped us to improve the paper.

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

References

1. Qiu, Y.; Li, B.; Shi, W.; Zhang, X. Visual Servo Tracking of Wheeled Mobile Robots with Unknown Extrinsic Parameters. *IEEE Trans. Ind. Electron.* **2019**, *66*, 8600–8609. [\[CrossRef\]](#)
2. Zhang, X.; Fang, Y.; Li, B.; Wang, J. Visual Servoing of Nonholonomic Mobile Robots with Uncalibrated Camera-to-Robot Parameters. *IEEE Trans. Ind. Electron.* **2017**, *64*, 390–400. [\[CrossRef\]](#)
3. Lin, Y.; Xing, K. Visual servo optimization stabilization of nonholonomic mobile robots based on control Lyapunov functions. *Meas. Control* **2020**, *53*, 1825–1831. [\[CrossRef\]](#)
4. Albekairi, M.; Mekki, H.; Kaaniche, K.; Yousef, A. An Innovative Collision-Free Image-Based Visual Servoing Method for Mobile Robot Navigation Based on the Path Planning in the Image Plan. *Sensors* **2023**, *23*, 9667. [\[CrossRef\]](#) [\[PubMed\]](#)
5. Al Arabi, A.; Tipu, R.S.; Bashar, M.R.; Barman, B.; Monicay, S.A.; Amin, M.A. Implementation of Low Cost Stereo Humanoid Adaptive Vision for 3D Positioning and Distance Measurement for Robotics Application with Self-Calibration. In Proceedings of the 2017 Asia Modelling Symposium (AMS), Kota Kinabalu, Malaysia, 4–6 December 2017; pp. 83–88. [\[CrossRef\]](#)
6. Pandey, S.K.; Kumar, L.; Kumar, G.; Kumar, A.; Singh, K.U.; Singh, T. Vision-Based Locomotion Control for Humanoid Robots: A Study on Vision-Guided Walking Strategies. In Proceedings of the 2023 International Conference on Computational Intelligence and Sustainable Engineering Solutions (CISES), Greater Noida, India, 28–30 April 2023; pp. 1036–1042. [\[CrossRef\]](#)
7. Zhang, X.; Fang, Y.; Zhang, X.; Jiang, J.; Chen, X. A Novel Geometric Hierarchical Approach for Dynamic Visual Servoing of Quadrotors. *IEEE Trans. Ind. Electron.* **2020**, *67*, 3840–3849. [\[CrossRef\]](#)
8. Lin, J.; Wang, Y.; Miao, Z.; Fan, S.; Wang, H. Robust Observer-Based Visual Servo Control for Quadrotors Tracking Unknown Moving Targets. *IEEE/ASME Trans. Mechatron.* **2023**, *28*, 1268–1279. [\[CrossRef\]](#)
9. Xie, W.F.; Li, Z.; Tu, X.W.; Perron, C. Switching Control of Image-Based Visual Servoing With Laser Pointer in Robotic Manufacturing Systems. *IEEE Trans. Ind. Electron.* **2008**, *56*, 520–529. [\[CrossRef\]](#)
10. Nguyen, D.T.H.; Nguyen, V.H. Application of Visual Servo in Tracking and Grasping Moving Target. In Proceedings of the 2022 International Conference on Control, Robotics and Informatics (ICCRI), Danang, Vietnam, 2–4 April 2022; pp. 83–87. [\[CrossRef\]](#)
11. Jiang, J.; Wang, Y.; Jiang, Y.; Xie, H.; Tan, H.; Zhang, H. A Robust Visual Servoing Controller for Anthropomorphic Manipulators with Field-of-View Constraints and Swivel-Angle Motion: Overcoming System Uncertainty and Improving Control Performance. *IEEE Robot. Autom. Mag.* **2022**, *29*, 104–114. [\[CrossRef\]](#)
12. Shu, T.; Gharaaty, S.; Xie, W.; Joubair, A.; Bonev, I.A. Dynamic Path Tracking of Industrial Robots with High Accuracy Using Photogrammetry Sensor. *IEEE/ASME Trans. Mechatron.* **2018**, *23*, 1159–1170. [\[CrossRef\]](#)
13. Lin, W.; Liu, C.; Guo, H.; Gao, H. Hybrid Visual-Ranging Servoing for Positioning Based on Image and Measurement Features. *IEEE Trans. Cybern.* **2023**, *53*, 4270–4279. [\[CrossRef\]](#)
14. Li, T.; Yu, J.; Qiu, Q.; Zhao, C. Hybrid Uncalibrated Visual Servoing Control of Harvesting Robots With RGB-D Cameras. *IEEE Trans. Ind. Electron.* **2023**, *70*, 2729–2738. [\[CrossRef\]](#)
15. He, S.; Xu, Y.; Guan, Y.; Li, D.; Xi, Y. Synthetic Robust Model Predictive Control With Input Mapping for Constrained Visual Servoing. *IEEE Trans. Ind. Electron.* **2023**, *70*, 9270–9280. [\[CrossRef\]](#)
16. Espiau, B.; Chaumette, F.; Rives, P. A New Approach to Visual Servoing in Robotics. In Proceedings of the Geometric Reasoning for Perception and Action, Workshop, Grenoble, France, 16–17 September 1991; Selected Papers.
17. Hutchinson, S.; Hager, G.D.; Corke, P.I. A tutorial on visual servo control. *IEEE Trans. Robot. Autom.* **1996**, *12*, 651–670. [\[CrossRef\]](#)
18. Chaumette, F.; Hutchinson, S. Visual servo control. I. Basic approaches. *IEEE Robot. Autom. Mag.* **2006**, *13*, 82–90. [\[CrossRef\]](#)
19. Chaumette, F. Potential problems of stability and convergence in image-based and position-based visual servoing. In *The Confluence of Vision and Control*; Springer: London, UK, 1998; pp. 66–78.
20. Chaumette, F. Image moments: A general and useful set of features for visual servoing. *IEEE Trans. Robot.* **2004**, *20*, 713–723. [\[CrossRef\]](#)
21. Tahri, O.; Chaumette, F. Point-based and region-based image moments for visual servoing of planar objects. *IEEE Trans. Robot.* **2005**, *21*, 1116–1127. [\[CrossRef\]](#)
22. Bateux, Q.; Marchand, E. Histograms-based visual servoing. *IEEE Robot. Autom. Lett.* **2016**, *2*, 80–87. [\[CrossRef\]](#)
23. Hafez, A.A.; Achar, S.; Jawahar, C. Visual servoing based on gaussian mixture models. In Proceedings of the 2008 IEEE International Conference on Robotics and Automation, Pasadena, CA, USA, 19–23 May 2008; IEEE: Piscataway, NJ, USA, 2008; pp. 3225–3230.
24. Crombez, N.; Mouaddib, E.M.; Caron, G.; Chaumette, F. Visual servoing with photometric gaussian mixtures as dense features. *IEEE Trans. Robot.* **2018**, *35*, 49–63. [\[CrossRef\]](#)
25. Ourak, M.; Tamadazte, B.; Lehmann, O.; Andreff, N. Wavelets-based 6 DOF visual servoing. In Proceedings of the 2016 IEEE International Conference on Robotics and Automation (ICRA), Stockholm, Sweden, 16–21 May 2016; IEEE: Piscataway, NJ, USA, 2016; pp. 3414–3419.
26. Liu, Y.H.; Wang, H.; Wang, C.; Lam, K.K. Uncalibrated visual servoing of robots using a depth-independent interaction matrix. *IEEE Trans. Robot.* **2006**, *22*, 804–817.
27. Anwar, A.; Lin, W.; Deng, X.; Qiu, J.; Gao, H. Quality Inspection of Remote Radio Units Using Depth-Free Image-Based Visual Servo with Acceleration Command. *IEEE Trans. Ind. Electron.* **2018**, *66*, 8214–8223. [\[CrossRef\]](#)
28. He, Z.; Wu, C.; Zhang, S.; Zhao, X. Moment-Based 2.5-D Visual Servoing for Textureless Planar Part Grasping. *IEEE Trans. Ind. Electron.* **2019**, *66*, 7821–7830. [\[CrossRef\]](#)

29. Bakthavatchalam, M.; Tahri, O.; Chaumette, F. A Direct Dense Visual Servoing Approach Using Photometric Moments. *IEEE Trans. Robot.* **2018**, *34*, 1226–1239. [[CrossRef](#)]
30. Li, S.T.; Ahmad, G.; Xie, W.F.; Gao, Y.B. An enhanced IBVS controller of a 6DOF manipulator using hybrid PD-SMC method. *International Journal of Control, Automation and Systems* **2018**, *16*, 844–855. [[CrossRef](#)]
31. Lazar, C.; Burlacu, A. Visual Servoing of Robot Manipulators Using Model-based Predictive Control. In Proceedings of the 2009 7th IEEE International Conference on Industrial Informatics, Cardiff, UK, 23–26 June 2009; pp. 690–695. [[CrossRef](#)]
32. Allibert, G.; Courtial, E.; Chaumette, F. Predictive Control for Constrained Image-Based Visual Servoing. *IEEE Trans. Robot.* **2010**, *26*, 933–939. [[CrossRef](#)]
33. Hajiloo, A.; Keshmiri, M.; Xie, W.F.; Wang, T.T. Robust Online Model Predictive Control for a Constrained Image-Based Visual Servoing. *IEEE Trans. Ind. Electron.* **2016**, *63*, 2242–2250. [[CrossRef](#)]
34. Bjerkeng, M.; Falco, P.; Natale, C.; Pettersen, K.Y. Discrete-time stability analysis of a control architecture for heterogeneous robotic systems. In Proceedings of the 2013 IEEE/RSJ International Conference on Intelligent Robots and Systems, Tokyo, Japan, 3–7 November 2013; pp. 4778–4783. [[CrossRef](#)]
35. Conticelli, F.; Allotta, B. Discrete-time robot visual feedback in 3D positioning tasks with depth adaptation. *IEEE/ASME Trans. Mechatron.* **2001**, *6*, 356–363. [[CrossRef](#)]
36. Bjerkeng, M.; Falco, P.; Natale, C. Stability Analysis of a Hierarchical Architecture for Discrete-Time Sensor-Based Control of Robotic Systems. *IEEE Trans. Robot.* **2014**, *30*, 745–753. [[CrossRef](#)]
37. He, S.; Xu, Y.; Li, D.; Xi, Y. Eye-in-Hand Visual Servoing Control of Robot Manipulators Based on an Input Mapping Method. *IEEE Trans. Control Syst. Technol.* **2023**, *31*, 402–409. [[CrossRef](#)]
38. Aicardi, M.; Caiti, A.; Cannata, G.; Casalino, G. Stability and robustness analysis of a two layered hierarchical architecture for the closed loop control of robots in the operational space. In Proceedings of the 1995 IEEE International Conference on Robotics and Automation, Nagoya, Japan, 21–27 May 1995; Volume 3, pp. 2771–2778. [[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.