Eye-in-Hand Visual Servoing Control of Robot Manipulators Based on an Input Mapping Method

Shaoying He, Yunwen Xu, Dewei Li, and Yugeng Xi, *Senior Member, IEEE*

***Abstract*—In image-based visual servoing (IBVS), parametric uncertainties tend to cause the model inaccuracy and limit the control performance. Considering these uncertainties can be embodied by the output–input data from the visual servoing system, this brief proposes an eye-in-hand visual servoing control (VSC) scheme based on the input mapping method, which directly utilizes the past output–input data to enhance the original feedback control law rather than identifying the model. The system with the input mapping method is proven to not only maintain the stability of the original VSC but also accelerate the convergent rate. The results of the experiments on a manipulator with an eye-in-hand camera demonstrate the superiority of our proposed method.**

***Index Terms*—Input mapping method, output–input data, parametric uncertainties, visual servo control.**

I. Introduction

IMAGE-BASED visual servoing (IBVS) is an approach to control motion of the robot by using visual feedback signals from the vision system. Due to the high reliability, the wide availability, and the low-cost feature, it has been widely used in different kinds of robots, including manipulators [1], inspection robots [2], mobile robots [3], and unmanned aerial vehicles [4]. The aim of this visual servoing control (VSC) is to minimize the image feature error $||e(k)|| = ||M(k) - M^*||$ by planning the robot motion, in which $M(k)$ and M^* are the current image feature and the desired image feature, respectively [5]. In common, the image feature includes point features [6], line features [7], moment features [8], and so on. The visual servoing model with different image features can be generally described by the image Jacobian matrix along with image parameters and system parameters.

To realize the aim of VSC, many different control methods have been developed based on the image Jacobian matrix. Espiau *et al.* [9] directly utilized the image Jacobian matrix from the desired image for control design. Although the design is simple and convenient, but a large model bias in the controller would cause a poor convergent rate. Hutchinson *et al.* [10] estimated the real-time image Jacobian matrix to construct the control law. The mean matrix between the estimated real-time image Jacobian matrix and the desired one was used to make the control law smooth in [11]. The visual servoing system under the above three control methods all can be convergent in experiments, while the control law with the more accurate image Jacobian matrix can do converge faster [12]. Thus, the more accurate image Jacobian matrix of the visual servoing model is, the higher performance is achieved. However, in a real system, the high-precision image Jacobian matrix is expensive to be acquired.

In application, the image Jacobian matrix depends not only on the camera intrinsic and extrinsic parameters, but also on depth parameters and pixel positions of feature points. For obtaining the accurate image Jacobian matrix, many methods have been developed to estimate the image Jacobian matrix or the uncertain parameters in this matrix. Kosmopoulos [13] proposed a robust off-line estimation method for small range workspace, but the control law was designed with only considering the fixed estimated image Jacobian matrix which brought conservatism to the control system. Jagersand *et al.* [14] employed the Broyden update rule to estimate the image Jacobian matrix online. Navarro-Alarcón *et al.* [15] further used the Broyden update rule to obtain a numerical estimation of Jacobian matrix in real-time and derived a dynamic-state feedback velocity control law to eliminate noise and uncertainty. Papanikolopoulos *et al.* [16] designed an estimation algorithm for estimating depth-related parameters. Piepmeier *et al.* [17] developed a dynamic Quasi-Newton method for estimating image Jacobian matrix. Shademan *et al.* [18] proposed a statistically robust M-estimator to reject the outliers in the past data. Xu *et al.* [19] used the different sensitive features to decouple the system and estimated the depths for all active features. An adaptive control approach was proposed in [20], which used the reinforcement learning method to adjust the parameter in estimation method and made the system more efficient and flexible than fixed parameter. The above methods improve the control performance by improving the accuracy of the identification model. But they separated the model identification from the

Manuscript received 15 October 2021; revised 19 January 2022; accepted 10 April 2022. Date of publication 18 May 2022; date of current version 28 December 2022. This work was supported in part by the National Key Research and Development Project under Grant 2018YFB1305902, in part by the National Science Foundation of China under Grant 61973214 and Grant 62103271, and in part by the China Postdoctoral Science Foundation under Grant 2021M692069. Recommended by Associate Editor C. Yang. *(Corresponding author: Yunwen Xu.)*

The authors are with the Department of Automation, Shanghai Jiao Tong University, Shanghai 200240, China (e-mail: heshaoying19941204@sjtu.edu.cn; willing419@sjtu.edu.cn; dwli@sjtu.edu.cn; ygxi@sjtu.edu.cn).

Color versions of one or more figures in this article are available at https://doi.org/10.1109/TCST.2022.3172571.

Digital Object Identifier 10.1109/TCST.2022.3172571

control design and the stability be guaranteed in the control process.

To solve the above problem, Liu *et al.* [21] proposed a depth-independent Jacobian matrix to replace the original image Jacobian matrix, which makes the estimated parameters appear linearly and identifies the accurate system model for better performance. In addition, the control law and the estimation update law are constructed by the Lyapunov theory and the method can be extended to the dynamic control. Later, Wang extended the above method to trajectory tracking with the point features, line features, and contour features [22]–[26]. To reduce the number of estimated parameters, Cheah *et al.* [27] proposed an update law to only estimate the uncertain depth information by depth-independent Jacobian matrix and design the control update law based on the Lyapunov theory. Liu *et al.* [28] unified this adaptive visual servoing with common image features. Liang *et al.* [29] unified the adaptive VSC in eye-in-hand/fixed configuration. The above adaptive methods can reduce the difficulty of model identification by linearizing the uncertain parameters and design the control law by the Lyapunov theory to guarantee the stability, but the effectiveness of these methods still depends on the accuracy of the identification model and the system optimality cannot be considered.

To overcome the above shortcoming, a VSC scheme equipped with the input mapping method (denoted as VSC-IM) is proposed in this brief. The concept of input mapping method, in which the linear combination of the past data is directly used to construct the control without the system identification, was proposed in [30] and [31]. In our proposed method, a moving window is used to collect the image error and the camera motion online. Then, the collected data are used to replace the part of uncertain model and the linear combination of the past control data is used to compensate the current control. Finally, the new control expression can be directly calculated by solving an optimization problem. In this way, the VSC can be directly improved with the data and the identification of image Jacobian matrix can be avoided.

The main contributions of our proposed method can be summarized as follows.

1) The past data including the image feature error and the motion of the camera can be directly used to compensate the original controller, which can use these data to replace the part model in design for reducing the influence of uncertainty.
2) This method can be proved to maintain the same stability as the original visual servoing system and obtain the faster convergent rate.

The rest is organized as follows. Section II briefly introduces the visual servoing system. Section III presents the design procedure of the VSC with input mapping method in detail. Subsequently, the theoretical analysis about the input mapping method in the visual servoing system is described in Section IV. Section V reports the experimental study. Section VI gives the final conclusion.

Notation: Denote $\boldsymbol{O}$ and $\boldsymbol{I}$ as the zero matrix and identity matrix with proper dimension, respectively. The largest singular value and the pseudo-inverse matrix of matrix $\boldsymbol{A} \in \mathbb{R}^{n\times m}$ are denoted by $\rho_{\max}(\boldsymbol{A})$ and $\boldsymbol{A}^{\dagger}$. And the $\mathbb{R}^n$ is equivalent to the $\mathbb{R}^{n\times 1}$.

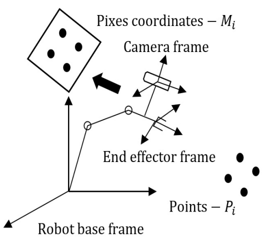

Fig. 1. Eye-in-hand setup for visual servoing.

II. Image-Based Visual Servoing

The aim of the IBVS is to make the manipulator drive the feature points to reach the desired feature coordinates in image. The input of this system is the camera motion V_c and the output is the feature coordinates M in the image. To simplify the problem, it is assumed that the camera has no distortion problem and the feature points are always in the field of view.

The illustration of eye-in-hand setup for visual servoing is in Fig. 1. The feature point in robot based frame and image plane is denoted as P_i and M_i ($i \in \{1, 2, \ldots, N\}$, $N \geq 3$). The transform matrix between the end-effector coordinate system and the robot base coordinate system is denoted as $\boldsymbol{T}_r^e(q) \in \mathbb{R}^{4\times 4}$, where the vector $q = [q_1, \ldots, q_n]^T \in \mathbb{R}^n$ denotes the manipulator joint position, n is the degree of freedom and $\boldsymbol{T}_r^e(q)$ can be obtained by the forward kinematic method [32]. The matrix between the camera coordinate system and the end-effector coordinate system is denoted as $\boldsymbol{T}_e^c \in \mathbb{R}^{4\times 4}$. The matrix between the camera coordinate system and the robot base coordinate system can be obtained by $\boldsymbol{T}_r^c(q) = \boldsymbol{T}_r^e(q)\boldsymbol{T}_e^c$.

A. Camera and Robot Manipulator Model

The feature points in the robot based coordinate system can be denoted as $P_i = [x_i, y_i, z_i]^T \in \mathbb{R}^3$, $i \in \{1, 2, \ldots, N\}$ ($N \geq 3$). Correspondingly, these feature points P_i in the camera frame are defined as $P_i^c = [x_i^c, y_i^c, z_i^c]^T \in \mathbb{R}^3$. The transformation relation between P_i and P_i^c can be defined as follows:

$$\Big[\big[P_i^c\big]^T, 1\Big]^T = \big(\boldsymbol{T}_r^c(q)\big)^{-1}\big[[P_i]^T, 1\big]^T. \tag{1}$$

The feature coordinates of P_i projected onto the image plane are set as $M_i = [u_i, v_i]^T \in \mathbb{R}^2$, which can be detected by the image processing method [33]. According to the camera projection model, the projection relationship between P_i^c and M_i are defined as follows [10]:

$$M_i = [u_i, v_i]^T = \boldsymbol{\psi}\big[\overline{x}_i, \overline{y}_i, 1\big]^T \tag{2}$$

where $[\overline{x}_i, \overline{y}_i]^T = [(x_i^c/z_i^c), {}^{y_i^c}/z_i^c]^T$ is the normalized coordinate and

$$\boldsymbol{\psi} = \begin{bmatrix} \alpha & 0 & u_0 \\ 0 & \beta & v_0 \end{bmatrix}$$

is the camera intrinsic matrix. The α and β are the scale factors of the image coordinate axis u and v.

The camera velocity is defined as $V_c = [v_c^T, w_c^T]^T \in \mathbb{R}^6$, where $v_c = [v_x, v_y, v_z]^T$ and $w_c = [w_x, w_y, w_z]^T$ are linear and angular velocities around the x-, y-, and z-axis, respectively. With the camera motion V_c, the motion of the feature coordinates M_i in image is described by

$$\dot{M}_i = \boldsymbol{J}_p^i V_c \tag{3}$$

where

$$\boldsymbol{J}_p^i = \begin{bmatrix} -\frac{\alpha}{z_i^c} & 0 & \frac{\bar{u}_i}{z_i^c} & \frac{\bar{u}_i \bar{v}_i}{\alpha} & -\frac{1}{\alpha} - \alpha \bar{u}_i^2 & \bar{v}_i \\ 0 & -\frac{\beta}{z_i^c} & \frac{\bar{v}_i}{z_i^c} & -\frac{1}{\beta} - \beta \bar{v}_i^2 & -\frac{\bar{u}_i \bar{v}_i}{\beta} & -\bar{u}_i \end{bmatrix}$$

and $\bar{u}_i = u_i - u_0$, $\bar{v}_i = v_i - v_0$. $\boldsymbol{J}_p^i \in \mathbb{R}^{2\times 6}$ denotes the image Jacobian matrix [34] of the feature coordinates M_i. And we can obtain

$$\dot{M} = \boldsymbol{L}_e V_c \tag{4}$$

where $M = [M_1^T, \ldots, M_N^T]^T \in \mathbb{R}^{2N}$ are the current feature and $\boldsymbol{L}_e = [\boldsymbol{J}_p^{1\,T}, \ldots, \boldsymbol{J}_p^{N\,T}]^T \in \mathbb{R}^{2N\times 6}$ is the image Jacobian matrices for all feature points. The aim of vision-based control schemes is to minimize the error e, which is defined by

$$e = M - M^* \tag{5}$$

where $M^* = [M_1^{*T}, \ldots, M_N^{*\,T}]^T \in \mathbb{R}^{2N}$ are the desired feature. With (4) and (5), the relation between camera velocity and the error can be obtained

$$\dot{e} = \boldsymbol{L}_e V_c. \tag{6}$$

The camera motion $V_c(k)$ is usually implemented by the manipulator. And the relation between the manipulator joint motion and the camera motion is modeled by

$$V_c = [v_c^T, w_c^T]^T = \boldsymbol{J}_r(q)\dot{q} \tag{7}$$

where $\boldsymbol{J}_r(q)$ is the manipulator geometric Jacobian which depends on the robot structure parameters. Then the manipulator can product the system input V_c for the visual servoing by $\dot{q} = (\boldsymbol{J}_r(q))^{\dagger} V_c$ to follow the controller command.

B. Visual Servoing Control

To reduce the error in (5), the discrete visual servoing feedback controller can be generally designed as

$$V_c(k) = -\boldsymbol{F}e(k) \tag{8}$$

where $e(k)$ is the image error and $V_c(k)$ is the corresponding control in the kth moment. With the control (8), the continuous model (6) can be discretized with sample time T

$$e(k+1) = e(k) + T\boldsymbol{L}_e V_c(k). \tag{9}$$

Then the Lyapunov function can be designed as

$$\mathcal{L}(k) = \|e(k)\|_{\boldsymbol{Q}}^2 = e(k)^T \boldsymbol{Q} e(k) \tag{10}$$

where $\boldsymbol{Q}$ is a proper positive matrix. And according to (9) and (10)

$$\Delta\mathcal{L}(k) = \|(I - T\boldsymbol{L_e}\boldsymbol{F})e(k)\|_{\boldsymbol{Q}}^2 - \|e(k)\|_{\boldsymbol{Q}}^2. \tag{11}$$

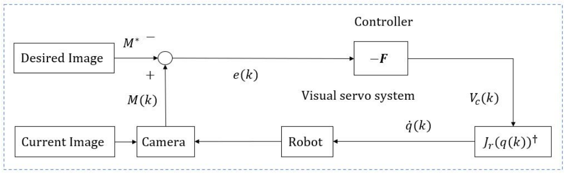

Fig. 2. Structure of visual servoing system.

Generally, the discrete system (9) is stable if and only if the following condition is ensured:

$$[\boldsymbol{I} - T\boldsymbol{L}_e\boldsymbol{F}]^T \boldsymbol{Q} [\boldsymbol{I} - T\boldsymbol{L}_e\boldsymbol{F}] < \boldsymbol{Q}. \tag{12}$$

There are already several methods that can make the system stable [10], [11], [35]. Here, we apply the method in [35] to the design of the basic feedback controller $\boldsymbol{F}$ by using the inverse of the desired image Jacobian matrix and a scale factor.

The work process of VSC is described as the Fig. 2. Firstly, the feedback law $\boldsymbol{F}$ can be obtained by the image Jacobian matrix $\boldsymbol{L}_e$ which describes the relationship between the image feature error and the camera motion. Then, the camera motion V_c can be calculated by the feedback law and the image error. After obtaining the V_c, the robot manipulator model in (7) is applied to get the manipulator joint motion $\dot{q}(k)$, and the robot manipulator implements the action to reduce the deviation from the desired image feature.

In the next section, we use the input mapping method to compensate the basic control law $\boldsymbol{F}$ for improving the performance of the original controller.

III. Visual Servoing Control With Input Mapping Method

In this section, we discuss the input mapping method first and then detail the implementation of VSC embedded with the input mapping method.

A. Input Mapping Method for Visual Servoing Control

For the small motion of the camera in a short time, the corresponding variation of image Jacobian matrix $\boldsymbol{L}_e$ is small [13]. In the visual servoing system (9), if the sample time T is small, the variation of the $T\boldsymbol{L}_e$ between the adjacent moments is very tiny and can be treated as constant in the adjacent moments. Therefore, the assumption about the discrete model $T\boldsymbol{L}_e$ is made as follows.

Assumption 1: The discrete model $T\boldsymbol{L}_e$ at the current moment is same as the past discrete model $T\boldsymbol{L}_e$ in the past l moments.

In application, we can set the sample time and the length l to be short for reducing the variation of the model. The error increment between two adjacent moments can be defined as follows:

$$\Delta e(k) = e(k+1) - e(k) = T\boldsymbol{L}_e V_c(k). \tag{13}$$

According to Assumption 1, the input–output data in the past l moments is available to describe the current system model. Thus, a moving window with a length l is used to collect the output/input data far from the current moment. Then, the formulas about (13) in the past l moments can be obtained by

$$\begin{aligned}\Delta e(k-1) &= T\boldsymbol{L}_e V_c(k-1)\\ &\vdots\\ \Delta e(k-l) &= T\boldsymbol{L}_e V_c(k-l).\end{aligned} \tag{14}$$

The vector form of (14) can be written as

$$\Delta\vec{e}(k) = T\boldsymbol{L}_e \vec{V}_c(k) \tag{15}$$

where $\Delta\vec{e}(k) = [\Delta e(k-1), \ldots, \Delta e(k-l)] \in \mathbb{R}^{2N\times l}$, $\vec{V}_c(k) = [V_c(k-1), \ldots, V_c(k-l)] \in \mathbb{R}^{6\times l}$ and $T\boldsymbol{L}_e \in \mathbb{R}^{2N\times 6}$. And the (15) describes the relation of past data. At the kth control moment, the past control sequence $\vec{V}_c(k)$ and their corresponding error increment $\Delta\vec{e}(k)$ can be easily obtained in the real system.

In here, we will use the input mapping method to improve system performance, in which the past data and their relation are used to replace the part of IBVS model as well as the past input data are directly used to compensate the original feedback control law.

To be specific, the control strategy for camera velocity $V_c(k)$ is composed of two terms. One is the input from the original feedback law and the other is the linear combination of the past data. That is,

$$V_c(k) = \hat{V}_c(k) + \vec{V}_c(k)\lambda(k) \tag{16}$$

where $\hat{V}_c(k)$ represents the original feedback law whose expression will be given later. $\vec{V}_c(k)\lambda(k)$ is the linear combination of the past control data and it is a subspace of $V_c(k)$. $\lambda(k) \in \mathbb{R}^l$ is the combination parameter vector, whose dimension depends on the length of l. The long data length l tends to add the computational burden for calculating $\lambda(k)$.

Considering the system error (9) with the new control (16), the error in the $k+1$ moment can be obtained as follows:

$$\begin{aligned}e(k+1) &= e(k) + T\boldsymbol{L}_e\Big[\hat{V}_c(k) + \vec{V}_c(k)\lambda(k)\Big]\\ &= e(k) + T\boldsymbol{L}_e\vec{V}_c(k)\lambda(k) + T\boldsymbol{L}_e\hat{V}_c(k).\end{aligned} \tag{17}$$

According to the mapping relation (15) of the input–output data, the parts $T\boldsymbol{L}_e\vec{V}_c(k)\lambda(k)$ of the model in the formula can be replaced as

$$\begin{aligned}e(k+1) &= e(k) + \Delta\vec{e}(k)\lambda(k) + T\boldsymbol{L}_e\hat{V}_c(k)\\ &= \hat{e}(k) + T\boldsymbol{L}_e\hat{V}_c(k)\end{aligned} \tag{18}$$

where $\hat{e}(k) \triangleq e(k) + \Delta\vec{e}(k)\lambda(k)$. To maintain the same stability, $\hat{V}_c(k)$ is designed as the similar structure of the original control method (8), and it can be written as $\hat{V}_c(k) = -\boldsymbol{F}\hat{e}(k)$.

The system aim is to reduce the error norm $||e(k+1)||^2_{\boldsymbol{Q}}$, then the optimization problem can be expressed as follows:

$$\begin{aligned}J &= \min_{\lambda(k)} \|e(k+1)\|^2_{\boldsymbol{Q}}\\ &= \min_{\lambda(k)} \Big\|e(k) + T\boldsymbol{L}_e\hat{V}_c(k) + \Delta\vec{e}(k)\lambda(k)\Big\|^2_{\boldsymbol{Q}}.\end{aligned} \tag{19}$$

And the solution of the above problem can be directly calculated as the following equation:

$$\lambda^*(k) = -\big[\Delta\vec{e}^T(k)\boldsymbol{Q}\Delta\vec{e}(k)\big]^\dagger \Delta\vec{e}^T(k)\boldsymbol{Q}\big[e(k) + T\boldsymbol{L}_e\hat{V}_c(k)\big]. \tag{20}$$

Algorithm 1 Visual Servoing Controller With Input Mapping Method (VSC-IM)

```
1: Choose the visual servoing method with a basic feedback
   control law $-\boldsymbol{F}e(k)$. Set initial condition, such as the
   past incremental error $\Delta\vec{e}(0) = 0$, combination parameter
   $\lambda(0) = 0$, the past control $\vec{V}_c(0) = 0$, learning rate $\eta$, data
   length $l$ and small scale $\varepsilon$.
2: At the begin of $k$th control moment, obtain $\lambda(k)$ by the
   formula (22). If the $\rho_{max}(\Delta\vec{e}^T(k)\boldsymbol{Q}\Delta\vec{e}(k)) < \varepsilon$, just set
   $\lambda(k) = 0$.
3: Measure the current error $e(k)$, obtain and store the
   current incremental error $\Delta e(k)$ into $\Delta\vec{e}(k)$. Use the
   past incremental error $\Delta\vec{e}(k)$ and past control $\vec{V}_c(k)$ to
   calculate the current control $V_c(k)$ by (23).
4: Get the joint speed $\dot{q}(k)$ of manipulator for $V_c(k)$ by (7),
   then implement $\dot{q}(k)$ on the robot.
5: At the end of $k$th control cycle, store the current control
   $V_c(k)$ into $\vec{V}_c(k)$. Let $k = k + 1$. Come back to Step.2.
```

Although we can get the rough model $\boldsymbol{L}_e$, the exact model of $\boldsymbol{L}_e$ is difficult to be obtained. Therefore, we remove this uncertain term $T\boldsymbol{L}_e\hat{V}_c(k)$ in (20) and simplify the solution as follows:

$$\lambda(k) = -\big(\Delta\vec{e}^T(k)\boldsymbol{Q}\Delta\vec{e}(k)\big)^\dagger \Delta\vec{e}^T(k)\boldsymbol{Q}e(k). \tag{21}$$

In addition, we add a learning rate $\eta \in (0,1)$ into the solution $\lambda(k)$ for adjusting the effect of the input mapping method. Thus, the final expression of $\lambda(k)$ is described as follows:

$$\lambda(k) = -\eta\big(\Delta\vec{e}^T(k)\boldsymbol{Q}\Delta\vec{e}(k)\big)^\dagger \Delta\vec{e}^T(k)\boldsymbol{Q}e(k). \tag{22}$$

Overall, the VSC equipped with input mapping can be constructed by

$$V_c(k) = -\boldsymbol{F}\big[e(k) + \Delta\vec{e}(k)\lambda(k)\big] + \vec{V}_c(k)\lambda(k). \tag{23}$$

However, there are an inverse operation $(\Delta\vec{e}^T(k)\boldsymbol{Q}\Delta\vec{e}(k))^\dagger$ in the expression of $\lambda(k)$. If $\Delta\vec{e}^T(k)\boldsymbol{Q}\Delta\vec{e}(k)$ is too small, it will make that the inverse operation in $\lambda(k)$ cannot be completed by the computer. To avoid this, when $\rho_{\max}(\Delta\vec{e}^T(k)\boldsymbol{Q}\Delta\vec{e}(k))$ is less than a very small scale ε, we directly make the $\lambda(k)$ to be zero. When $\lambda(k) = 0$, the system control (23) degrades into the original system control (8) without the input mapping method and the system stability will not be affected.

B. Implementation of Visual Servoing Control With the Input Mapping Method

According to the above description, the input mapping method steps for VSC are given in Algorithm 1 and one can get the flowchart of how the VSC works with input mapping module as shown in Fig. 3. Firstly, the camera catches the current feature coordinate $M(k)$. Then, the image feature error $e(k)$ can be calculated by $M(k) - M^*$. Next, the camera velocity $V_c(k)$ can be constructed by the proposed VSC-IM method. Finally, the manipulator joint speed $\dot{q}$ can

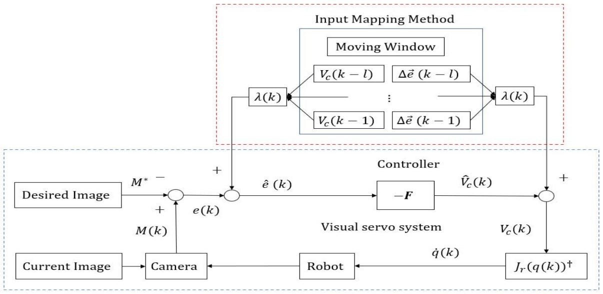

Fig. 3. Visual servoing system with input mapping method.

be calculated according to (7) and executed on the robot to support the camera velocity.

Different with other visual servoing methods, the key contribution of the VSC-IM is to compensate the basic visual servoing feedback law $\boldsymbol{F}$ by the linear combination of past data. In next section, we will prove the VSC-IM can maintain the stability and improve the convergent rate of the basic visual servoing feedback law.

IV. Theoretical Analysis

In this section, the influence of the VSC-IM will be analyzed. It can be proved that the stability of VSC under the input mapping method is still maintained and the convergent rate of the system can be accelerated.

A. Stability of System

Theorem 1: Consider the visual servoing system, the basic VSC law $\boldsymbol{F}$ with input mapping method (23), when the control law $\boldsymbol{F}$ keeps the system stable, the controller equipped with the input mapping method (16) also keeps the system stable.

Proof: To simplify the expression of $e(k)$, define the term $\Delta\vec{e}(k)\lambda(k)$ as follows:

$$e'(k) \triangleq -\Delta\vec{e}(k)\lambda(k). \tag{24}$$

Then, the $e(k)$ can be divided into the following parts:

$$e(k) = \hat{e}(k) + e'(k). \tag{25}$$

Substituting the $\lambda(k)$ in (22) into $\hat{e}(k)$ and $e'(k)$ and the expression of $\hat{e}(k)$ and $e'(k)$ can be shown as follows:

$$e'(k) = \eta\Delta\vec{e}(k)\big[\Delta\vec{e}^T(k)\boldsymbol{Q}\Delta\vec{e}(k)\big]^\dagger\Delta\vec{e}^T(k)\boldsymbol{Q}e(k) \tag{26}$$

$$\hat{e}(k) = e(k) - e'(k). \tag{27}$$

According to (26) and (27), the matrix product between $\hat{e}$ and $e'(k)$ with $\boldsymbol{Q}$ can be obtained as follows:

$$\begin{aligned}\hat{e}^T(k)\boldsymbol{Q}e'(k) &= \big(e(k)-e'(k)\big)^T\boldsymbol{Q}e'(k)\\ &= \eta(1-\eta)e^T(k)\boldsymbol{Q}\Delta\vec{e}(k)\big[\Delta\vec{e}^T(k)\boldsymbol{Q}\Delta\vec{e}(k)\big]^\dagger\\ &\quad\times\Delta\vec{e}^T(k)\boldsymbol{Q}e(k) \geq 0.\end{aligned} \tag{28}$$

According to the error model (17) and input mapping control (23), we can define the error in $(k+1)$th control moment as follows:

$$\begin{aligned}e(k+1)_{\text{VSC-IM}} &= e(k) + \Delta\vec{e}(k)\lambda(k) + \boldsymbol{T}\boldsymbol{L}_e\hat{V}_c(k)\\ &= \hat{e}(k) - \boldsymbol{T}\boldsymbol{L}_e\boldsymbol{F}\hat{e}(k).\end{aligned} \tag{29}$$

The feedback control $\boldsymbol{F}$ can satisfy the stable condition (12), then we can have

$$\left\|e(k+1)_{\text{VSC-IM}}\right\|_{\boldsymbol{Q}}^2 = \|(\boldsymbol{I}-\boldsymbol{T}\boldsymbol{L}_e\boldsymbol{F})\hat{e}(k)\|_{\boldsymbol{Q}}^2 < \|\hat{e}(k)\|_{\boldsymbol{Q}}^2. \tag{30}$$

According to condition $\hat{e}^T(k)\boldsymbol{Q}e'(k) \geq 0$, the following relation can be obtained:

$$\begin{aligned}\|e(k)\|_{\boldsymbol{Q}}^2 &= \|\hat{e}(k)\|_{\boldsymbol{Q}}^2 + \left\|e'(k)\right\|_{\boldsymbol{Q}}^2 + 2\hat{e}^T(k)\boldsymbol{Q}e'(k)\\ &\geq \|\hat{e}(k)\|_{\boldsymbol{Q}}^2.\end{aligned} \tag{31}$$

Therefore, the $||e(k+1)_{\text{VSC-IM}}||_{\boldsymbol{Q}}^2 < ||e(k)||_{\boldsymbol{Q}}^2$ can be proved, and VSC-IM is stable as the basic control system.

B. Convergent Rate Analysis of System

Theorem 2: Consider the visual servoing system, the basic feedback control law (8) with input mapping method (23), when VSC-IM has the same situation as VSC in the kth control moment, the VSC-IM can converge more faster than the basic method, i.e., $||e(k+1)_{\text{VSC}}||_{\boldsymbol{Q}}^2 \geq ||e(k+1)_{\text{VSC-IM}}||_{\boldsymbol{Q}}^2$.

Proof: With the same precondition, the error of the VSC and VSC-IM in the $(k+1)$th moment under two kinds of control (13) and (23) can be written as follows:

$$e(k+1)_{\text{VSC}} = e(k) + \boldsymbol{T}\boldsymbol{L}_e[-\boldsymbol{F}e(k)], \tag{32}$$

$$e(k+1)_{\text{VSC-IM}} = e(k) + \boldsymbol{T}\boldsymbol{L}_e\Big[-\boldsymbol{F}\hat{e}(k) + \vec{V}_c(k)\lambda(k)\Big]. \tag{33}$$

Consider the expressions of $e(k)$ and $\lambda(k)$, the above expressions (32) and (33) can be transformed as follows:

$$e(k+1)_{\text{VSC}} = [\boldsymbol{I}-\boldsymbol{T}\boldsymbol{L}_e\boldsymbol{F}]\big[\hat{e}(k)+e'(k)\big] \tag{34}$$

$$e(k+1)_{\text{VSC-IM}} = [\boldsymbol{I}-\boldsymbol{T}\boldsymbol{L}_e\boldsymbol{F}]\hat{e}(k). \tag{35}$$

Therefore, the error norm of VSC and VSC-IM can be written as follows:

$$\begin{aligned}\|e(k+1)_{\text{VSC}}\|_{\boldsymbol{Q}}^2 &= \|[\boldsymbol{I}-\boldsymbol{T}\boldsymbol{L}_e\boldsymbol{F}]\hat{e}(k)\|_{\boldsymbol{Q}}^2\\ &\quad+\left\|[\boldsymbol{I}-\boldsymbol{T}\boldsymbol{L}_e\boldsymbol{F}]e'(k)\right\|_{\boldsymbol{Q}}^2\\ &\quad+2[[\boldsymbol{I}-\boldsymbol{T}\boldsymbol{L}_e\boldsymbol{F}]\hat{e}(k)]\boldsymbol{Q}\\ &\quad+\big[[\boldsymbol{I}-\boldsymbol{T}\boldsymbol{L}_e\boldsymbol{F}]e'(k)\big]^T\end{aligned} \tag{36}$$

$$\left\|e(k+1)_{\text{VSC-IM}}\right\|_{\boldsymbol{Q}}^2 = \|[\boldsymbol{I}-\boldsymbol{T}\boldsymbol{L}_e\boldsymbol{F}(k)]\hat{e}(k)\|_{\boldsymbol{Q}}^2 \tag{37}$$

where $||[\boldsymbol{I}-\boldsymbol{T}\boldsymbol{L}_e\boldsymbol{F}]e'(k)||_{\boldsymbol{Q}}^2 \geq 0$ and $[[\boldsymbol{I}-\boldsymbol{T}\boldsymbol{L}_e\boldsymbol{F}]\hat{e}(k)]\boldsymbol{Q}[[\boldsymbol{I}-\boldsymbol{T}\boldsymbol{L}_e\boldsymbol{F}]e'(k)]^T \geq 0$. Thus, it is obvious that the follows can be obtained:

$$\|e(k+1)_{\text{VSC}}\|_{\boldsymbol{Q}}^2 \geq \left\|e(k+1)_{\text{VSC-IM}}\right\|_{\boldsymbol{Q}}^2. \tag{38}$$

Remark 1: From (36) to (37), it is obvious that the error norms of the VSC-IM is always less than VSC. The $||\overrightarrow{e}(k+1)_{\text{VSC}}||_{\boldsymbol{Q}}^2 = ||\overrightarrow{e}(k+1)_{\text{VSC-IM}}||_{\boldsymbol{Q}}^2$ can be satisfied if and only if the $e'(k)$ is zero. And according to (24), it can be obtained that $e'(k) = 0$ means that the $\Delta\vec{e}(k) = 0$ or $e(k) = 0$. When the increment error or the error become to zero, it means that the control process is over. Thus, at the end of the control process the two methods will have an equivalent performance, while in other situations $||\overrightarrow{e}(k+1)_{\text{VSC-IM}}||_{\boldsymbol{Q}}^2$ made

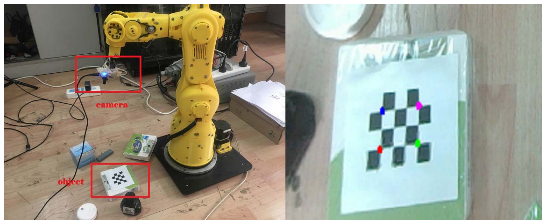

Fig. 4. Operation platform and the feature points of visual servoing system.

TABLE I

D-H MODEL OF MANIPULATOR

i	α_{i-1}	a_{i-1}(m)	θ_{i-1}	d_{i-1}(m)
1	$0°$	0	θ_1	$L_1 = 0.242$
2	$-90°$	0	$\theta_2 - 90°$	0
3	$0°$	$L_2 = 0.225$	$\theta_3 + 90°$	0
4	$90°$	0	θ_4	$L_3 = 0.229$
5	$90°$	0	$-\theta_5$	0
6	$-90°$	0	θ_6	$L_4 = 0.050$

by VSC with the input mapping method is always smaller than $||\overrightarrow{e}(k+1)_{\mathrm{VSC}}||^2_{\boldsymbol{Q}}$. Thus, when the basic controller equipped with the input mapping module, the error norms can be less and the system convergent rate is faster than before.

In conclusion, according to Theorems 1 and 2, if the basic method is stable, i.e., satisfy the stable condition (12), the VSC-IM can keep the same stability as the basic controller and accelerate its convergent rate.

V. EXPERIMENTS

In this section, we have designed several comparison experiments to validate the superiority of the proposed method. The experimental environment is installed as shown in Fig. 4. The robot system chosen in the experiments is a six degree of freedom manipulator and the monocular camera is installed at the end. And we select corner points in the checkerboard as the feature points, where the selected feature points are colored.

A. Experiment Parameter Setting

The corresponding D-H parameters of the manipulator are listed in Table I. The repeated allocation precision of the robot is 0.05° and the image pixel size is 1080 × 960. Due to the limitation of the robot accuracy and the image pixel size, the camera intrinsic matrix and the extrinsic matrix between the camera and the end-effector are roughly calibrated as follows:

$$\psi = \begin{bmatrix} 860.3766 & 0 & 686.1657 \\ 0 & 860.4384 & 526.7917 \end{bmatrix} \tag{39}$$

$$T_e^c = \begin{bmatrix} 0.01998 & 0.9997 & 0.0099 & -0.0401 \\ -0.9998 & 0.0199 & 0.0001 & -0.0121 \\ 0.0001 & -0.0098 & 0.9999 & -0.0452 \\ 0 & 0 & 0 & 1 \end{bmatrix}. \tag{40}$$

These inaccurate matrices along with the unknown camera motion cause the uncertain variation of the image Jacobian matrix in the control process. Then our proposed method and other adaptive are used to adjust the control or identify the model for better performance.

The conventional VSC in [35] with the rough model is selected as the based control law and the input mapping method is directly used to reduce the influence of model inaccuracy and improve the system performance. To prove the superiority of the input mapping method, the other adaptive methods including visual servoing control with the quasi-Newton method (VSC-QN) [17] and uncalibrated visual servoing with a depth-independent Jacobian (VSC-DIJ) [26] are selected for the comparison.

TABLE II

INITIAL (I) AND DESIRED (D) LOCATION OF FEATURE POINTS IN PIXEL

		Point1	Point2	Point3	Point4
first	I	(388,178)	(388,45)	(255,178)	(255,45)
	D	(427,522)	(408,436)	(333,553)	(318,465)
second	I	(500,826)	(457,696)	(364,860)	(318,731)
	D	(693,592)	(685,436)	(532,602)	(512,447)

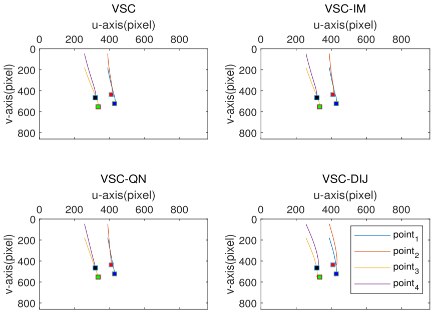

Fig. 5. Trajectories of the different methods for the first task.

For other parameters, we set the control cycle $T = 0.04$ s. The learning rate is set as $\eta = 0.01$. And we set the minimal pixel error as $\varepsilon = 5$ and the length as $l = 2$. The depth information is set 0.5 m and the initial and desired locations of the features are given in Table II. The basic feedback law $\boldsymbol{F}$ in [35] is constructed by the desired coordinate and depth information and the positive matrix $\boldsymbol{Q}$ is set as $0.1\boldsymbol{I}$. The feedback law $\boldsymbol{F}$ is set as the initial feedback law for the other adaptive methods in experiment and the other known parameters are also set same.

B. Experiment Results and Analysis

The trajectories in image for two groups of the different desired positions are placed in Figs. 5 and 6, which show that all the visual servoing method can steer the feature points to the desired positions and the desired features are marked with the squares. And the results of the input mapping method have been created in 3-D graphs as Fig. 7 where the red coordinate system, purple coordinate system, and blue curve are initial position, desired position and trajectory. The image errors under these different control methods are shown in Fig. 8 and converge to zero from the image feature initial errors. The values of λ along time are shown in Fig. 9, which shows that after a period of time when the system error is enough small ($<\varepsilon$), the λ is set to be zero and the input mapping module is closed as described in Algorithm 1. In the experiments, the maximum computation time of our proposed method in one cycle is 2.18 ms which is much less than the control cycle. Finally, an HD video demonstrating our method better convergence performance than other method in the real environments can be found at https://youtu.be/1fU8uH5Y3LI.

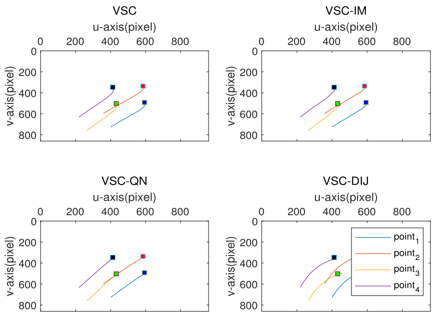

Fig. 6. Trajectories of the different methods for the second task.

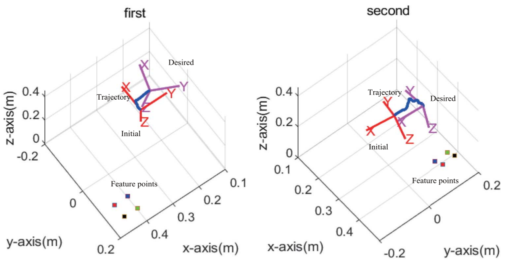

Fig. 7. Trajectories of the input mapping methods in 3-D for the first and second tasks.

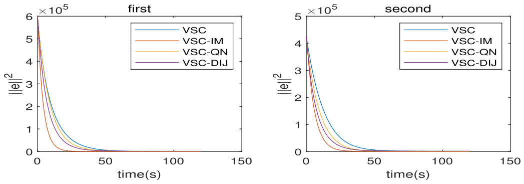

Fig. 8. $||e||^2$ of the different methods for the first and second tasks.

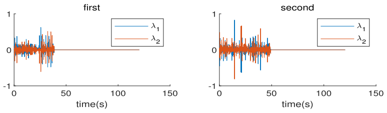

Fig. 9. Values of $\lambda(k)$ along time for the first and second tasks.

In experiments, the conventional VSC is constructed by the rough model. The VSC can eliminate the system error by the feedback mechanism, but this way extends the process of system convergence and affect the system performance. For the VSC-QN and VSC-DIJ, the kinds of adaptive algorithm can get the more accuracy model by collecting the past input–output data and identify model, then adjust the adaptive control law for acquiring better convergent rate. And for the VSC-IM, we use the input–output data and their relation to replace the part of the uncertain model in design so that the influence of the inaccuracy model can be weaken. Thus, it can be seen that the VSC-IM, VSC-QN and VSC-DIJ can acquire better convergence performance than VSC in Fig. 8.

These kinds of adaptive methods including VSC-QN and VSC-DIJ need identify the model and get the accuracy model, then the performance can be improved. Our proposed method can directly use the past data and their relation to replace the part of model and compensate the control, so that the model identification can be avoided. In addition, the linear combination of the past control data can be directly used to compensate the original control, and the linear combination can be calculated by solving an optimization problem, so that the control performance can be improved in the beginning. Therefore, the VSC-IM can converge better than other adaptive methods in Fig. 8.

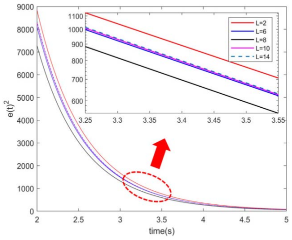

Fig. 10. $||e||^2$ of the different data length.

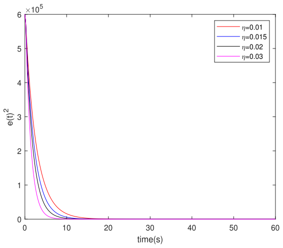

Fig. 11. $||e||^2$ of the different η.

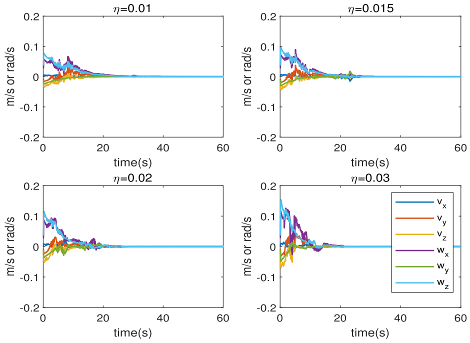

Fig. 12. Camera motion of the different η.

C. Effect of the Data Length l and the Learn Rate η

In VSC-IM, there are two parameters needed to be calibrated: the data length l and the learn rate η. And we will discuss the selection of the two parameters.

We taking the first experiment with different parameters for example again. The l is selected as $(2, 6, 8, 10, 14)$, respectively. The $||e||^2$ of the VSC-IM with the different data length between two and five seconds is shown in Fig. 10. The zoom-in picture of results is shown on the top-right corner. It can be found that the convergent rate increases with the increase of data length before $l = 8$, while the convergent rate decreases with the increase of data length after $l = 8$. This is because in short period of time, the model changes a little and the collected data can describe the model more accurate. But with the increase of l, the system stores the past data produced by the past model which is different with the current model, which may have a negative effect on the system and hurt the control performance. Moreover, the long data length l will also add the computational burden for calculating $\lambda(k)$. Thus, the data length should not be selected too long.

The $||e||^2$ of the VSC-IM with different learning rate is shown in Fig. 11 and the control of the camera motion V_c is shown in Fig. 12, where the learning rate η is selected as $(0.01, 0.015, 0.02, 0.03)$, respectively. The learning rate η represents the proportion of input mapping module in the original control. The larger the learning rate is, the stronger the role of the input mapping module will be, and the greater the system performance will be improved. Thus, the convergent rate increases with the increase of the learning rate η in Fig. 11. But it also can be found that the large learning rate η tends to increase the control V_c from Fig. 12. The oversize camera motion would be beyond manipulator physical constraints or out of camera view. Thus, the learn rate η should not be selected too large in the real application.

VI. Conclusion

In this brief, to eliminate the influence of model uncertainty we propose a visual servoing control frame equipped with the input mapping method. This design utilizes the past data to update the control input directly, reduce the negative effects of model inaccuracy and improve system performance without the system identification. Meanwhile, it is proven that the stability of the system with input mapping method is maintained and the convergence rate is faster than the original system. At last, the experiments on an eye-in-hand visual servoing system show that the input mapping design can obtain better performance.

References

[1] Z. Gong, B. Tao, C. Qiu, Z. Yin, and H. Ding, “Trajectory planning with shortest path for modified uncalibrated visual servoing based on projective homography,” *IEEE Trans. Autom. Sci. Eng.*, vol. 17, no. 2, pp. 1076–1083, Apr. 2020.

[2] C. Zang and K. Hashimoto, “A flexible visual inspection system combining pose estimation and visual servo approaches,” in *Proc. IEEE Int. Conf. Robot. Autom.*, May 2012, pp. 1304–1309.

[3] H. Shi, M. Xu, and K.-S. Hwang, “A fuzzy adaptive approach to decoupled visual servoing for a wheeled mobile robot,” *IEEE Trans. Fuzzy Syst.*, vol. 28, no. 12, pp. 3229–3243, Dec. 2020.

[4] M. A. Rafique and A. F. Lynch, “Output-feedback image-based visual servoing for multirotor unmanned aerial vehicle line following,” *IEEE Trans. Aerosp. Electron. Syst.*, vol. 56, no. 4, pp. 3182–3196, Aug. 2020.

[5] J. Bingxi, S. Liu, K. Zhang, and J. Chen, “Survey on robot visual servo control: Vision system and control strategies,” *Acta Autom. Sinica*, vol. 41, no. 5, pp. 861–873, 2015.

[6] M. Kazemi, K. K. Gupta, and M. Mehrandezh, “Randomized kinodynamic planning for robust visual servoing,” *IEEE Trans. Robot.*, vol. 29, no. 5, pp. 1197–1211, Oct. 2013.

[7] L. J.-Y. Xu De, “Determination for interactive matrix of line feature,” *Acta Automatica Sinica*, vol. 41, no. 10, pp. 1762–1771, 2015.

[8] M. Bakthavatchalam, O. Tahri, and F. Chaumette, “A direct dense visual servoing approach using photometric moments,” *IEEE Trans. Robot.*, vol. 34, no. 5, pp. 1226–1239, Oct. 2018.

[9] B. Espiau, F. Chaumette, and P. Rives, “A new approach to visual servoing in robotics,” *IEEE Trans. Robot. Automat.*, vol. 8, no. 3, pp. 313–326, Jun. 1992.

[10] S. Hutchinson, G. D. Hager, and P. I. Corke, “A tutorial on visual servo control,” *IEEE Trans. Robot. Autom.*, vol. 12, no. 5, pp. 651–670, Oct. 1996.

[11] O. Tahri and Y. Mezouar, “On visual servoing based on efficient second order minimization,” *Robot. Auto. Syst.*, vol. 58, no. 5, pp. 712–719, May 2010.

[12] X. De, “A tutorial for monocular visual servoing,” *Acta Automatica Sinica*, vol. 44, no. 10, pp. 1729–1746, 2018.

[13] D. I. Kosmopoulos, “Robust Jacobian matrix estimation for image-based visual servoing,” *Robot. Comput.-Integr. Manuf.*, vol. 27, no. 1, pp. 82–87, Feb. 2011.

[14] M. Jagersand, O. Fuentes, and R. Nelson, “Experimental evaluation of uncalibrated visual servoing for precision manipulation,” in *Proc. Int. Conf. Robot. Automat.*, vol. 4, Apr. 1997, pp. 2874–2880.

[15] D. Navarro-Alarcon, Y.-H. Liu, J. G. Romero, and P. Li, “Model-free visually servoed deformation control of elastic objects by robot manipulators,” *IEEE Trans. Robot.*, vol. 29, no. 6, pp. 1457–1468, Dec. 2013.

[16] N. P. Papanikolopoulos, B. J. Nelson, and P. K. Khosla, “Six degree-of-freedom hand/eye visual tracking with uncertain parameters,” *IEEE Trans. Robot. Automat.*, vol. 11, no. 5, pp. 725–732, Oct. 1995.

[17] J. A. Piepmeier, G. V. McMurray, and H. Lipkin, “Uncalibrated dynamic visual servoing,” *IEEE Trans. Robot. Autom.*, vol. 20, no. 1, pp. 143–147, Feb. 2004.

[18] A. Shademan, A.-M. Farahmand, and M. Jägersand, “Robust Jacobian estimation for uncalibrated visual servoing,” in *Proc. IEEE Int. Conf. Robot. Autom.*, May 2010, pp. 5564–5569.

[19] D. Xu, J. Lu, P. Wang, Z. Zhang, and Z. Liang, “Partially decoupled image-based visual servoing using different sensitive features,” *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 47, no. 8, pp. 2233–2243, Aug. 2017.

[20] H. Shi, G. Sun, Y. Wang, and K.-S. Hwang, “Adaptive image-based visual servoing with temporary loss of the visual signal,” *IEEE Trans. Ind. Informat.*, vol. 15, no. 4, pp. 1956–1965, Apr. 2018.

[21] Y.-H. Liu, H. Wang, C. Wang, and K. K. Lam, “Uncalibrated visual servoing of robots using a depth-independent interaction matrix,” *IEEE Trans. Robot.*, vol. 22, no. 4, pp. 804–817, Aug. 2006.

[22] H. Wang, Y. Liu, and D. Zhou, “Dynamic visual tracking for manipulators using an uncalibrated fixed camera,” *IEEE Trans. Robot.*, vol. 23, no. 3, pp. 610–617, Jun. 2007.

[23] H. Wang, Y. H. Liu, and D. Zhou, “Adaptive visual servoing using point and line features with an uncalibrated eye-in-hand camera,” *IEEE Trans. Robot.*, vol. 24, no. 4, pp. 843–857, Aug. 2008.

[24] H. Wang, Y.-H. Liu, and W. Chen, “Uncalibrated visual tracking control without visual velocity,” *IEEE Trans. Control Syst. Technol.*, vol. 18, no. 6, pp. 1359–1370, Nov. 2010.

[25] H. Wang, Y.-H. Liu, W. Chen, and Z. Wang, “A new approach to dynamic eye-in-hand visual tracking using nonlinear observers,” *IEEE/ASME Trans. Mechatronics*, vol. 16, no. 2, pp. 387–394, Apr. 2011.

[26] H. Wang, B. Yang, J. Wang, X. Liang, W. Chen, and Y.-H. Liu, “Adaptive visual servoing of contour features,” *IEEE/ASME Trans. Mechatronics*, vol. 23, no. 2, pp. 811–822, Apr. 2018.

[27] C. C. Cheah, C. Liu, and J. J. E. Slotine, “Adaptive Jacobian vision based control for robots with uncertain depth information,” *Automatica*, vol. 46, no. 7, pp. 1228–1233, Jul. 2010.

[28] Y.-H. Liu, H. Wang, W. Chen, and D. Zhou, “Adaptive visual servoing using common image features with unknown geometric parameters,” *Automatica*, vol. 49, no. 8, pp. 2453–2460, 2013.

[29] X. Liang, H. Wang, Y. Liu, W. Chen, and J. Zhao, “A unified design method for adaptive visual tracking control of robots with eye-in-hand/fixed camera configuration,” *Automatica*, vol. 59, pp. 97–105, Sep. 2015.

[30] L. Yang, J. Lu, Y. Xu, D. Li, and Y. Xi, “Constrained robust model predictive control embedded with a new data-driven technique,” *IET Control Theory Appl.*, vol. 14, no. 16, pp. 2395–2405, 2020.

[31] D. Li *et al.*, “Synthesis of ILC–MPC controller with data-driven approach for constrained batch processes,” *IEEE Trans. Ind. Electron.*, vol. 67, no. 4, pp. 3116–3125, Apr. 2020.

[32] B. Siciliano, L. Sciavicco, L. Villani, and G. Oriolo, *Robotics: Modelling, Planning and Control*. Springer, 2010.

[33] Z. Zhang, “A flexible new technique for camera calibration,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 22, no. 11, pp. 1330–1334, Dec. 2000.

[34] C. P. Bechlioulis, S. Heshmati-alamdari, G. C. Karras, and K. J. Kyriakopoulos, “Robust image-based visual servoing with prescribed performance under field of view constraints,” *IEEE Trans. Robot.*, vol. 35, no. 4, pp. 1063–1070, Aug. 2019.

[35] F. Chaumette and S. Hutchinson, “Visual servo control, Part I: Basic approaches,” *IEEE Robot. Autom. Mag.*, vol. 13, no. 4, pp. 82–90, Dec. 2006.