

Article

## Robust Kalman Filtering Cooperated Elman Neural Network Learning for Vision-Sensing-Based Robotic Manipulation with Global Stability

Xungao Zhong, Xunyu Zhong \* and Xiafu Peng

Department of Automation, Xiamen University, South Siming Road, Xiamen 361005, China;  
E-Mails: zhongxungao@163.com (X.Z.); pengxiafu@126.com (X.P.)

\* Author to whom correspondence should be addressed; E-Mail: zhongxunyu@xmu.edu.cn;  
Tel.: +86-137-2088-5429; Fax: +86-592-258-0005.

Received: 10 August 2013; in revised form: 2 September 2013 / Accepted: 2 September 2013 /  
Published: 8 October 2013

---

**Abstract:** In this paper, a global-state-space visual servoing scheme is proposed for uncalibrated model-independent robotic manipulation. The scheme is based on robust Kalman filtering (KF), in conjunction with Elman neural network (ENN) learning techniques. The global map relationship between the vision space and the robotic workspace is learned using an ENN. This learned mapping is shown to be an approximate estimate of the Jacobian in global space. In the testing phase, the desired Jacobian is arrived at using a robust KF to improve the ENN learning result so as to achieve robotic precise convergence of the desired pose. Meanwhile, the ENN weights are updated (re-trained) using a new input-output data pair vector (obtained from the KF cycle) to ensure robot global stability manipulation. Thus, our method, without requiring either camera or model parameters, avoids the corrupted performances caused by camera calibration and modeling errors. To demonstrate the proposed scheme's performance, various simulation and experimental results have been presented using a six-degree-of-freedom robotic manipulator with eye-in-hand configurations.

**Keywords:** visual servoing; dynamic Jacobian estimation; Kalman filtering; Elman neural network; global-state-space

---

## 1. Introduction

Visual sensors integrated with robotic manipulators can be increasingly beneficial for robotic perception and behavioral flexibility in unstructured environments [1]; such sensors have caught much attention, and have applications in all walks of life [2–5].

Vision-based robotic manipulation depends mainly on visual information feedback to control the positioning or motioning of a manipulator [6]. There are two general categories of manipulation: position-based visual servoing (PBVS) and image-based visual servoing (IBVS). There is also a hybrid of the two techniques [7]. These different categories arise from differential definitions of visual feature and feedback control information.

The PBVS vision usage provides information to regulate the end-effector pose (relative to the object in the Cartesian space). Owing to its characteristic of global asymptotic stability, this method is suitable for most industrial robotic manipulators [8]. However, the servoing task is inevitably related to the tedious affine model which is associated with world coordinates, camera frame, and robot base coordinates. Consequently, PBVS is more sensitive, with respect to the camera and object modeling errors [9], and to the possibility of disappearing image features [10–12].

In IBVS, there is direct control of the feature points on the image plane for robotic manipulation, and the image Jacobian matrix is used to describe the differential relation between the image error and the end-effector pose [13]; such direct control has simplified PBVS computation. However, IBVS has its intrinsic drawbacks. Consequently, modified IBVS method have recently mainly focused on: (1) the dynamic visual servo control law, which was proposed to improve poor dynamic performances due to either low vision acquisition frequency or to the time latency processing system [14]; (2) image frame path planning as an effective solution for making camera calibration and robotic kinematics modeling errors more robust [15–17] (such optimization techniques are selected for minimizing both end-effector trajectories in Cartesian space and features trajectories on image planes [18]); and (3) redundant and cooperative 2D visual servo systems [5] and intelligent hybrid schemes [19], which solve typical problems, such as features extraction errors and task singularities, *etc.*

With respect to modified methods, the presented image Jacobian matrix requires depth information and camera calibration parameters (as in traditional IBVS methodologies); such requirements inevitably lead to task singularities, thus making it difficult to ensure the stabilized convergence of a desired target [20]. The Jacobian identification is considered to be a dynamic parameter estimation problem. Solutions to this issue are broadly grouped into two categories.

One category comprises of online estimation techniques (e.g., the famous Broyden-based method and its modified variants) [21,22]. However, these estimation techniques may actually depend on the system configurations and the tasks to be accomplished. Moreover, these techniques are ill-suited for dealing with system environmental noises. In [23], an M-estimator was applied to the Jacobian estimation task, which does not require the model and system parameters. The first literature discussing the Kalman-Bucy filter (KBF) on state space for online Jacobian estimation is proposed in [24]. Those methods have the same properties as recursive least squares-based estimation processing's. However, the Jacobian estimation task may involve singularity instability in global space; failure to have end-effector positioning in cases of large displacement between the initial and desired poses is also a risk.

Another solution to this estimation problem involves machine learning techniques [25,26], which are based on biologically inspired approaches, such as neural networking. These approaches enable the machine to learn the kinematic relationship of manipulators during action-perception cycles [27,28]. While in IBVS learning techniques, the Jacobian is presented by a trained nonlinear network [25], this approach approximates convergence to the desired position; reaching levels of high servoing precision is difficult because network training is an incomplete convergence process and is sensitive to training samples.

In this paper, the discussion will focus on non-parameter Jacobian matrix estimation problems. A new global-state-space IBVS scheme, which associates KF and ENN learning techniques, is proposed for uncalibrated model-independent robotic manipulation that has robust stability in global-state-space, where the image features are constrained on the camera field-of-view (FOV). The proposed scheme is different from traditional PBVS methods; IBVS methods possess the merits of not needing calibration of either camera parameters or the robotic kinematics model. Moreover, for IBVS methods, Jacobian dynamic estimation does not require depth information. The main contributions of the paper are as follows:

- (1) Jacobian online identification problems were solved by introducing state-space infrastructure, which has been incorporated into robust KF techniques. The traditional KF is a minimum-variance state estimator for linear dynamic systems with Gaussian white noises sequence. In most practices, however, the observation noise is compound and the noises are statistically correlated noises (as to being simple white noise sequences). Therefore, we have derived a robust KF estimator with multiple orders of noise for the Jacobian online estimation task.
- (2) The KF is sensitive with respect to the initial robotic state and the initial noises' statistical characteristics (*i.e.*, a small perturbation of noise characteristics to dynamic modeling will lead to serious positioning error). For this problem, the ENN was adopted as a global estimator for Jacobian learning, and then the global map relationship between vision space and the robot workspace is represented by the ENN. Notice that this learning method is an approximation of the Jacobian matrix, with respect to the end-effector desired pose in scale motion space; the approximation error present in the offline learning is due to the limited size of the training data.
- (3) In the online testing phase, the precise positioning problem is solved by using a robust KF to improve ENN learning so as to achieve robotic convergence to the desired pose. After, the ENN's weights are updated through re-training using a new input-output data pair vector, which is obtained from the KF cycle to ensure robotic global stability manipulation. Finally, we have designed a novel global-state-space IBVS framework associated with robust KF cooperated ENN learning. In our finding, the image Jacobian matrix is estimated without accounting for camera calibration and modeling error. Our servoing system performs robustly despite outside noises and system destabilization.

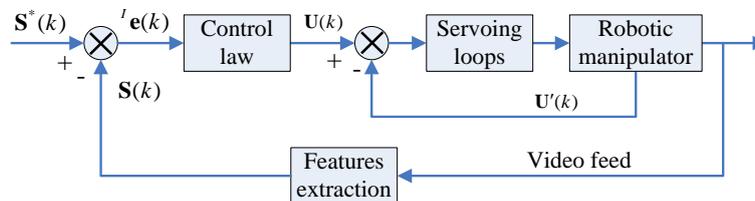
The paper is organized as follows: a general description of visual servoing (VS) for uncalibrated model-independent robotic manipulation is presented in the next section. The Jacobian estimation with KF is presented in the Section 3; in that section, we have also derived a robust KF with statistically correlated noises. The ENN for global Jacobian learning and a novel global-state-space IBVS scheme

are presented in Section 4. The simulation and experimental results are discussed in Section 5. Section 6 presents our conclusions.

## 2. Visual Servoing to Uncalibrated Model-Independent Robotic Manipulation

In this section, we assume that we have a robotic manipulator,  $\mathbf{R}$ , that lacks a kinematics model, and where the camera parameters are uncalibrated. As described in Figure 1, uncalibrated model-independent visual servoing (VS) aims to drive the robotic end-effector from the current image feature ( $\mathbf{S}(k)$ ,  $\mathbf{S}(k) \in \mathfrak{R}^m$ ) to the desired image feature ( $\mathbf{S}^*(k)$ ,  $\mathbf{S}^*(k) \in \mathfrak{R}^m$ ) by using the control variable ( $\mathbf{U}(k)$ ,  $\mathbf{U}(k) \in \mathfrak{R}^n$ ). Generally, a simple local motion of the end-effector will result in the nonlinear complex change of many features on the image plane. The solution for describing the change of features and the robot motion, through the adoption of an image Jacobian matrix, was originally proposed in [13].

**Figure 1.** The general structure of uncalibrated model-independent VS system.



The task error  ${}^{\chi}\mathbf{e}(k)$  in the  $n$ -dimensional workspace  $\chi \in \mathfrak{R}^n$  can be described as:

$${}^{\chi}\mathbf{e}(k) = {}^{\chi}\mathbf{P}_E(k) - {}^{\chi}\mathbf{P}_T(k): \quad {}^{\chi}\mathbf{P}_E(k) \times {}^{\chi}\mathbf{P}_T(k) \rightarrow \mathfrak{R}^n \tag{1}$$

where  ${}^{\chi}\mathbf{P}_E(k)$  and  ${}^{\chi}\mathbf{P}_T(k)$  represent the end-effector and target pose, respectively.

The  $m$ -dimensional feature vector on image frame  $\mathbf{I}$  is used for task manipulation; image error  ${}^I\mathbf{e}(k)$  is given by:

$${}^I\mathbf{e}(k) = {}^I\mathbf{h}_E({}^{\chi}\mathbf{P}_E(k)) - {}^I\mathbf{h}_T({}^{\chi}\mathbf{P}_T(k)) = \mathbf{S}(k) - \mathbf{S}^*(k): \quad {}^I\mathbf{h}_E \times {}^I\mathbf{h}_T \rightarrow \mathfrak{R}^m \tag{2}$$

Where  ${}^I\mathbf{h}_E$ ,  ${}^I\mathbf{h}_T$  belong to the end-effector and target; these variables result from the projection of  ${}^{\chi}\mathbf{P}_E(k)$ ,  ${}^{\chi}\mathbf{P}_T(k)$  to the image plane, respectively. The Equation (2) is a decision function, when  ${}^I\mathbf{e}(k) = 0$ , that meaning is actually equivalent to robot achieving the task manipulation, *i.e.*,  ${}^{\chi}\mathbf{e}(k) = 0$ .

The association of  ${}^I\mathbf{e}(k) \in \mathfrak{R}^m$  with the differential changes of the end-effector pose  ${}^{\chi}\mathbf{P}_E(k) \in \mathfrak{R}^n$  is achieved by assuming linearity through the image Jacobian matrix  $\mathbf{J}(k)$ ; this is done through the equation [13]:

$${}^I\dot{\mathbf{e}}(k) = \mathbf{J}(k) \cdot {}^{\chi}\dot{\mathbf{P}}_E(k) \Leftrightarrow \dot{\mathbf{S}}(k) = \mathbf{J}(k) \cdot {}^{\chi}\dot{\mathbf{P}}_E(k) \tag{3}$$

where  $\mathbf{J}(k)$  is defined as:

$$\mathbf{J}(k) = \frac{\partial \mathbf{S}(k)}{\partial {}^{\chi}\mathbf{P}_E(k)} = \begin{bmatrix} \frac{\partial s^1(k)}{\partial {}^{\chi}p_E^1(k)} & \dots & \frac{\partial s^1(k)}{\partial {}^{\chi}p_E^n(k)} \\ \dots & \dots & \dots \\ \frac{\partial s^m(k)}{\partial {}^{\chi}p_E^1(k)} & \dots & \frac{\partial s^m(k)}{\partial {}^{\chi}p_E^n(k)} \end{bmatrix}_{m \times n} \tag{4}$$

### 3. Jacobian Online Estimation by Introduction in State-Space

#### 3.1. The State Equation and Observation Equation

Robot attached visual sensors can be considered an instrumental system whose state vector is formed by concatenations of the row and column elements of the image Jacobian matrix  $\mathbf{J}(k)$  [24], *i.e.*,

$$\mathbf{X}(k) = \left[ \frac{\partial s^1(k)}{\partial^x \mathbf{P}_E(k)} \quad \frac{\partial s^2(k)}{\partial^x \mathbf{P}_E(k)} \quad \dots \quad \frac{\partial s^m(k)}{\partial^x \mathbf{P}_E(k)} \right]^T \quad (5)$$

where  $\frac{\partial s^i(k)}{\partial^x \mathbf{P}_E(k)} = \left[ \frac{\partial s^i(k)}{\partial^x p_E^1} \quad \frac{\partial s^i(k)}{\partial^x p_E^2} \quad \dots \quad \frac{\partial s^i(k)}{\partial^x p_E^n} \right]^T$  ( $i = 1, 2, 3, \dots, n$ ).

The robotic state-space model is a linear discrete-time dynamical system according to:

$$\mathbf{X}(k+1) = \mathbf{X}(k) + \mathbf{W}(k) \quad (6)$$

$$\mathbf{Z}(k) = \mathbf{H}(k)\mathbf{X}(k) + \mathbf{V}(k) \quad (7)$$

where  $\mathbf{X}(k) \in \mathfrak{R}^{m \times n}$  is the state vector,  $\mathbf{W}(k) \in \mathfrak{R}^{m \times n}$  is the model noise with zero mean, and variances is  $\mathbf{Q}(k)$ ,  $\mathbf{V}(k) \in \mathfrak{R}^{m \times 1}$  is the observation noise vector,  $\mathbf{Z}(k) \in \mathfrak{R}^{m \times 1}$  is the output vector, given by:

$$\mathbf{Z}(k) = \mathbf{S}(k+1) - \mathbf{S}(k) = \mathbf{J}(k)^x \dot{\mathbf{P}}_E(k)^T \quad (8)$$

Thus, the observation matrix  $\mathbf{H}(k)$  is

$$\mathbf{H}(k) = \begin{bmatrix} {}^x \dot{p}_E(k)^T & & 0 \\ & \dots & \\ 0 & & {}^x \dot{p}_E(k)^T \end{bmatrix}_{m \times mn} \quad (9)$$

The Kalman-Bucy filter (KBF) [29] is a minimum-variance state estimator for linear dynamic systems with Gaussian white noise sequences; the KBF has been used for real-time state estimation [30], as well as for solving the Jacobina matrix online estimation problem [24]. In most practices, however, the observation noises ( $\mathbf{V}(k)$ ) may be compound, and statistically correlated with model noises ( $\mathbf{W}(k)$ ), rather than being simple white noise. In the next section, we provide a robust Kalman filtering with statistically correlated noises for image Jacobian estimation task.

#### 3.2. Robust KF for Jacobian Estimation

For the universal environment, the observation noise vector ( $\mathbf{V}(k)$ ) that meets the Markov chain model, given by:

$$\mathbf{V}_{(k)} = \boldsymbol{\Psi}_{(k-1)} \mathbf{V}_{(k-1)} + \boldsymbol{\eta}_{(k-1)} \quad (10)$$

where,  $\boldsymbol{\Psi}_{(k-1)}$  is the coefficient matrix,  $\boldsymbol{\eta}_{(k)}$  is the Gaussian white noise sequence with zero mean, and the variance is  $\mathbf{R}_{(k)}$ .

Based upon Equations (6), (7) and (10), we obtain the observation value of system state at  $k + 1$  time, as follows:

$$\begin{aligned}
\mathbf{Z}^{(k+1)} &= \mathbf{H}_{(k+1)}\mathbf{X}^{(k+1)} + \mathbf{V}^{(k+1)} \\
&= \mathbf{H}_{(k+1)}[\mathbf{X}^{(k)} + \mathbf{W}^{(k)}] + \boldsymbol{\Psi}_{(k)}\mathbf{V}^{(k)} + \boldsymbol{\eta}^{(k)} \\
&= (\mathbf{H}_{(k+1)} - \boldsymbol{\Psi}_{(k)}\mathbf{H}_{(k)})\mathbf{X}^{(k)} + \boldsymbol{\Psi}_{(k)}\mathbf{Z}^{(k)} + \mathbf{H}_{(k+1)}\mathbf{W}^{(k)} + \boldsymbol{\eta}^{(k)}
\end{aligned} \tag{11}$$

Equation (11) can be rewritten as:

$$\underbrace{\mathbf{Z}^{(k+1)} - \boldsymbol{\Psi}_{(k)}\mathbf{Z}^{(k)}}_{\mathbf{Z}^*_{(k)}} = \underbrace{[\mathbf{H}_{(k+1)} - \boldsymbol{\Psi}_{(k)}\mathbf{H}_{(k)}]}_{\mathbf{H}^*_{(k)}}\mathbf{X}^{(k)} + \underbrace{\mathbf{H}_{(k+1)}\mathbf{W}^{(k)} + \boldsymbol{\eta}^{(k)}}_{\mathbf{V}^*_{(k)}} \tag{12}$$

Equation (12) can be considered as an equivalent observation equation to replace Equation (7); the statistic characteristics of  $\mathbf{W}^{(k)}$  and  $\mathbf{V}^*_{(k)}$  are as follows:

$$\begin{cases} E[\mathbf{V}^*_{(k)}\mathbf{V}^*_{(j)}{}^T] = \underbrace{(\mathbf{H}_{(k+1)}\mathbf{Q}_{(k)}\mathbf{H}_{(k+1)}^T + \mathbf{R}_{(k)})}_{\mathbf{R}^*_{(k)}}\delta_{kj} \\ E[\mathbf{W}^{(k)}\mathbf{V}^*_{(j)}{}^T] = \underbrace{(\mathbf{Q}_{(k)}\mathbf{H}_{(k+1)}^T)}_{\mathbf{S}_{(k)}}\delta_{kj} \end{cases}, \quad \delta_{kj} = \begin{cases} 1 & k = j \\ 0 & k \neq j \end{cases} \tag{13}$$

Equation (13) shows that the observation noise ( $\mathbf{V}^*_{(k)}$ ) is a Gaussian white noise sequence with zero mean, the variance is  $\mathbf{R}^*_{(k)}$ .  $\mathbf{V}^*_{(k)}$  is statistically correlated with the model noise  $\mathbf{W}^{(k)}$ , and the cross-covariance matrix is  $\mathbf{S}_{(k)}$ . In this case, the application of traditional KF methods was limited. To solve this problem, we introduced a filtering-revised-vector  $\boldsymbol{\rho}^{(k)}$ , such that state Equation (6) can be written as:

$$\begin{aligned}
\mathbf{X}^{(k)} &= \mathbf{X}^{(k-1)} + \mathbf{W}^{(k-1)} + \boldsymbol{\rho}^{(k-1)}\underbrace{(\mathbf{Z}^*_{(k-1)} - \mathbf{H}^*_{(k-1)}\mathbf{X}^{(k-1)} - \mathbf{V}^*_{(k-1)})}_{=0} \\
&= \underbrace{(\mathbf{E} - \boldsymbol{\rho}^{(k-1)}\mathbf{H}^*_{(k-1)})}_{\boldsymbol{\Phi}^{(k-1)}}\mathbf{X}^{(k-1)} + \boldsymbol{\rho}^{(k-1)}\mathbf{Z}^*_{(k-1)} + \underbrace{\mathbf{W}^{(k-1)} - \boldsymbol{\rho}^{(k-1)}\mathbf{V}^*_{(k-1)}}_{\mathbf{W}^*_{(k-1)}}
\end{aligned} \tag{14}$$

In order to eliminate the relevance between  $\mathbf{W}^*_{(k)}$  and  $\mathbf{V}^*_{(k)}$ , we let  $E[\mathbf{W}^*_{(k)}\mathbf{V}^*_{(j)}{}^T] = 0$ , i.e.:

$$E[\mathbf{W}^*_{(k)}\mathbf{V}^*_{(j)}{}^T] - \boldsymbol{\rho}^{(k)}E[\mathbf{V}^*_{(k)}\mathbf{V}^*_{(j)}{}^T] = (\mathbf{S}_{(k)} - \boldsymbol{\rho}^{(k)}\mathbf{R}^*_{(k)})\cdot\delta_{kj} = 0 \tag{15}$$

Thus, the filtering-revised-vector is:

$$\boldsymbol{\rho}^{(k)} = \mathbf{S}_{(k)}\mathbf{R}^*_{(k)}{}^{-1} \tag{16}$$

Equations (12) and (14) can be considered as a equivalent observation equation and system state equation, respectively. According to the optimal estimation theory [29], the robust Kalman filtering model of statistically correlated noises can be derived as follows:

- (1) Estimation equations: the one-step state estimation equation and the variance matrix of estimation error are given by:

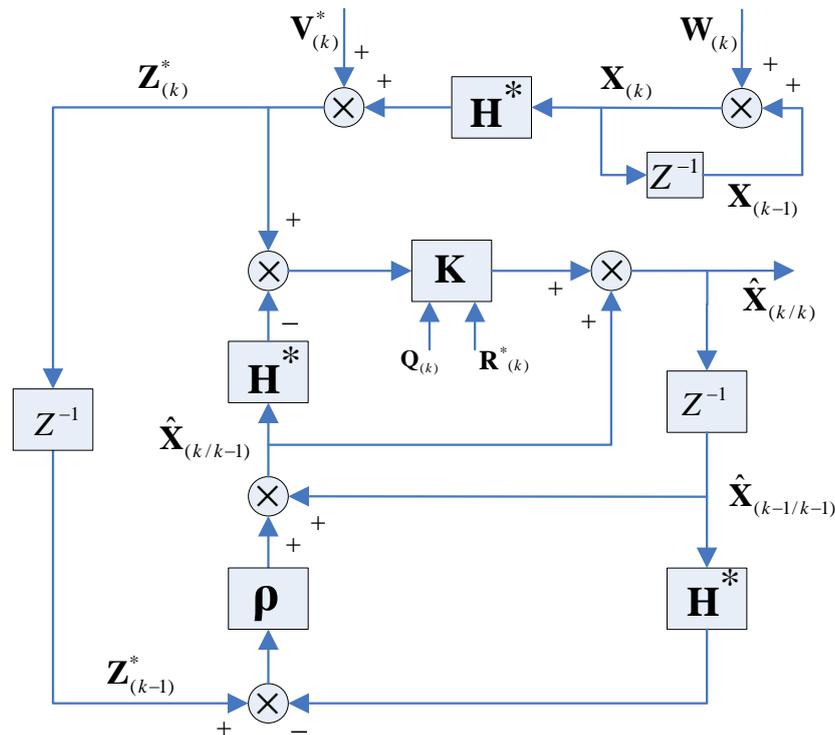
$$\begin{cases} \hat{\mathbf{X}}_{(k/k-1)} = \hat{\mathbf{X}}_{(k-1/k-1)} + \boldsymbol{\rho}^{(k-1)}[\mathbf{Z}^*_{(k-1)} - \mathbf{H}^*_{(k-1)}\hat{\mathbf{X}}_{(k-1/k-1)}] \\ \mathbf{P}_{(k/k-1)} = E[(\mathbf{X}^{(k)} - \hat{\mathbf{X}}_{(k/k-1)})(\mathbf{X}^{(k)} - \hat{\mathbf{X}}_{(k/k-1)})^T] \\ \quad = [\mathbf{E} - \boldsymbol{\rho}^{(k-1)}\mathbf{H}^*_{(k-1)}]\mathbf{P}_{(k-1/k-1)}[\mathbf{E} - \boldsymbol{\rho}^{(k-1)}\mathbf{H}^*_{(k-1)}]^T + \mathbf{Q}_{(k-1)} - \boldsymbol{\rho}^{(k-1)}\mathbf{S}_{(k-1)}^T \end{cases} \tag{17}$$

- (2) Updating equations: the state filtering equation, filtering gain, and variance matrix for filtering error are as follows:

$$\begin{cases} \hat{\mathbf{X}}_{(k/k)} = \hat{\mathbf{X}}_{(k/k-1)} + \mathbf{K}_{(k)}(\mathbf{Z}_{(k)}^* - \mathbf{H}_{(k)}^* \hat{\mathbf{X}}_{(k/k-1)}) \\ \mathbf{K}_{(k)} = \mathbf{P}_{(k/k-1)} \mathbf{H}_{(k)}^{*T} [\mathbf{H}_{(k)}^* \mathbf{P}_{(k/k-1)} \mathbf{H}_{(k)}^{*T} + \mathbf{R}_{(k)}^*]^{-1} \\ \mathbf{P}_{(k/k)} = E[(\mathbf{X}_{(k)} - \hat{\mathbf{X}}_{(k/k)})(\mathbf{X}_{(k)} - \hat{\mathbf{X}}_{(k/k)})^T] \\ \quad = [\mathbf{I} - \mathbf{K}_{(k)} \mathbf{H}_{(k)}^*] \mathbf{P}_{(k/k-1)} \end{cases} \quad (18)$$

As shown in Figure 2, a robust KF has two basic processes, which are the estimation and the observation updates. The estimation process uses prior-state information and a state model to estimate the current state. The observation update uses the discrepancy between the estimated value and the observed value.

**Figure 2.** The structure of a robust KF for Jacobian matrix estimation. We introduce a filtering-revise-vector  $\boldsymbol{\rho}_{(k)}$  to improve the robustness of the filter’s performance with respect to universal dynamic noises.



#### 4. A Global-State-Space IBVS Scheme for Robotic Manipulation

In this section, a novel global-state-space IBVS method is proposed, which attaches the robust KF Cooperate with Elman neural network learning techniques, so as to enable robot stability manipulation in global-state-space.

##### 4.1. Elman Neural Network for Jacobian Approximate Learning

The original Elman neural network (ENN) was proposed by Elman in 1990 [31]. ENN is a typical dynamic recurrent network, with particular context nodes (other than the common input layer): the hidden and output layers. ENN’s biggest advantage is that its context nodes may be used to memorize the previous activations of the hidden nodes; this makes ENN applicable in the fields of dynamic

system identification and prediction control. In this paper, as shown in Figure 3, the framework of feedback ENN with its three layer structure is selected; the framework consists of the input vector  $\mathbf{S}(k)$ , output vector  $\mathbf{J}(k)$ , connection weights  $\mathbf{W}^{Li}$  (where  $i = 1, 2, \dots, 5$ ), the nodal activation functions,  $h(\cdot)$ ,  $g(\cdot)$ ,  $f(\cdot)$ , and the thresholds,  $\alpha_i, \alpha_j, \alpha_k$ , which correspond to the input layer, hidden layer, and output layer, respectively. The mathematical of the ENN is described as follows [32]:

Input layers:

$$\begin{cases} \mathbf{O}^I(k) = h(\mathbf{W}^{L1}\mathbf{I}^C(k) + \mathbf{W}^{L2}\mathbf{S}(k) - \alpha_i) \\ \mathbf{I}^C(k) = \mathbf{O}^I(k-1) \end{cases} \quad (19)$$

Hidden layers:

$$\begin{cases} \mathbf{O}^H(k) = g(\mathbf{W}^{L3}\mathbf{O}^I(k) + \mathbf{W}^{L4}\mathbf{I}^H(k) - \alpha_j) \\ \mathbf{I}^H(k) = \mathbf{O}^H(k-1) \end{cases} \quad (20)$$

Output layers:

$$\mathbf{J}(k) = f(\mathbf{W}^{L5}\mathbf{O}^H(k) - \alpha_k) \quad (21)$$

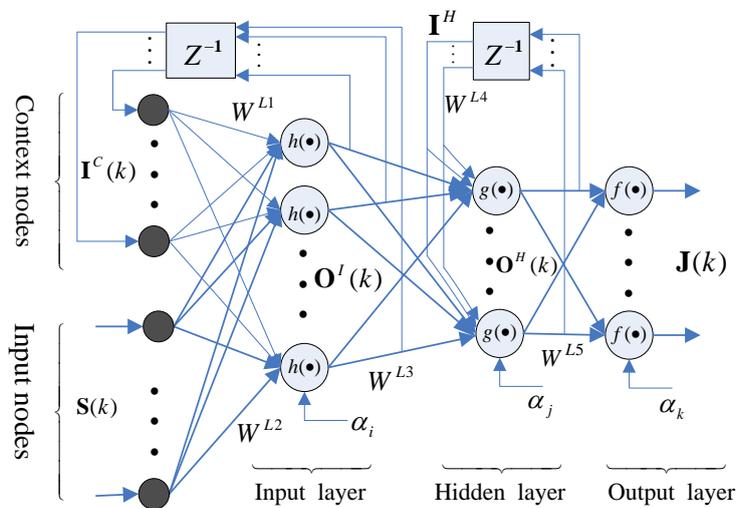
where the activation function of input layer  $h(x)$  and hidden layer  $g(x)$  are taken as a sigmoidal function:

$$h(x) = g(x) = \frac{1}{1 + e^{-x}} \quad (22)$$

The output layer activation function  $f(x)$  is taken as a linear function, such that Equation (21) is written as:

$$\mathbf{J}(k) = \mathbf{W}^{L5}\mathbf{O}^H(k) - \alpha_k \quad (23)$$

**Figure 3.** The framework of a feedback Elman neural network with its three layer structure.



The learning algorithm is applied to determine the neural network structure. The goal of training is to obtain the convergence of the connection weights  $\mathbf{W}^{Li}$  ( $i = 1, 2, \dots, 5$ ) of the input layer to hidden layer, and the hidden layer to output layer. Where 600 samples are used to train the ENN, the input

samples to the ENN are the image feature vector ( $\mathbf{S}'(k) = [\mathbf{s}_1(k), \mathbf{s}_2(k), \dots, \mathbf{s}_{600}(k)]_{8 \times 600}$ ), and the output samples of the network are the Jacobian matrix ( $\mathbf{J}'(k) = [\mathbf{x}_1(k), \mathbf{x}_2(k), \dots, \mathbf{x}_{600}(k)]_{48 \times 600}$ ). The learning laws with the gradient descent method [32], and the training result (approximately at epoch 107), reach the best validation performances, and the minimum sum squared error (MSE) is 1.3. It is seen that the test output of the network is not very close to the training output, which means that the actual end-effector positioning is approximately close to the desired target; this is due to the approximation error present in offline learning.

#### 4.2. Global-State-Space IBVS Scheme

The schematic of the proposed global-state-space IBVS is shown in Figure 4. The goal is to extract image features by vision sensor to estimate, online, the desired Jacobian  $\hat{\mathbf{J}}(k)$ , so as to drive the end-effector to reach the desired pose, as employed by a control law. The scheme consists of the following steps:

(i) First, the Jacobian for the global mapping relationship between the vision space and the robot workspace is approximated using the ENN. As mentioned in last section, it is not necessary to have a very low approximation error during offline training. But in the testing phase, the supposed desired Jacobian  $\hat{\mathbf{J}}(k)$ , with respect to the desired end-effector pose, is estimated by the robust KF (as shown in the dashed box in Figure 4). The ENN's weights were undated and use the gradient descent algorithm to minimize output error  $e(k)$  for a given input image features vector  $\mathbf{S}(k-1)$ .

$$E(k) = \frac{1}{2} e^T(k)e(k) \quad (24)$$

where  $e(k) = (\hat{\mathbf{J}}(k) - \mathbf{J}'(k))$ . The update law of the connecting weight is given by:

$$w(k+1) = w(k) + \eta e(k) \frac{\partial J'(k)}{\partial w} \quad (25)$$

where  $\eta$  is the learning rate.

(ii) The initial state of the robot is very important to robot stability manipulation. The authors in [33] use a fuzzy neural network to provide this initial joint guess. In [34], the authors start from the current robot configuration and in [27], a KSOM-based sub-clustering network is used to provide an initial joint guess to the inversion algorithm. For the initial Jacobian guess, the common method involves introducing the robotic probe moving at the neighborhood of its initial position  $n$  times  $\partial p^{j-n+1} \dots \partial p^j$ , and observing the corresponding features displacements  $\partial f^{j-n-1} \dots \partial f^j$  in the image frame. The initial Jacobian matrix could then be obtained reasonably through [24]:

$$\hat{\mathbf{J}}(0) = [\partial f^{j-n-1} \dots \partial f^j] [\partial p^{j-n+1} \dots \partial p^j]^T \quad (26)$$

In contrast, in this paper, at the initial time  $k = 0$ , the initial image features  $\mathbf{S}(0)$  captured by the vision sensor are used as an input for ENN. After, the ENN outputs an initial Jacobian guess  $\mathbf{J}(0)$ , which can be considered the initial state  $\mathbf{X}(0)$  of the robot. Therefore, our scheme is more flexible since it avoids introducing complex probe motions.

(iii) The robust KF estimator (mentioned in Section 3.2) is used to estimate the desired Jacobian  $\hat{\mathbf{J}}(k)$  through the following steps:

- Step 1 robot state initialization as Step (ii).
- Step 2 At  $k-1$  time, the approximated Jacobian  $\mathbf{J}(k-1)$  is calculated using Equation (23).
- Step 3  $\mathbf{X}(k-1) \leftarrow \mathbf{J}(k-1)$ , robot state updating using state Equation (14).
- Step 4 Output vector  $\mathbf{Z}(k)$  is obtained using observation Equation (12).
- Step 5 The best state estimate  $\hat{\mathbf{X}}(k)$  is obtained using the robust KF Equations (17) and (18).
- Step 6  $\hat{\mathbf{J}}(k) \leftarrow \hat{\mathbf{X}}(k)$ .
- Step 7 ENN weights update through Equation (25).
- Step 8  $k-1 \leftarrow k$ , go to Step 2.

(iv) The control law should be employed to drive the robot from its present pose to its desired pose. The image error, as shown in Equation (2) at the time instant  $k$ , is rewritten as:

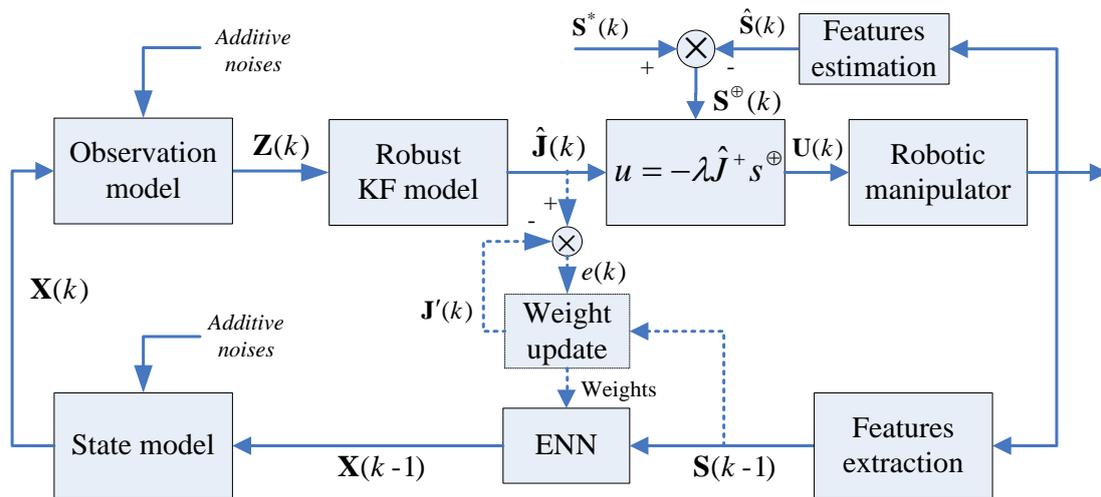
$$\mathbf{S}^{\oplus}(k) = \mathbf{S}^*(k) - \hat{\mathbf{S}}(k) \tag{27}$$

where  $\mathbf{S}^*(k)$  is the desired image feature, and  $\hat{\mathbf{S}}(k) = \mathbf{S}(k-1) + (\mathbf{S}(k-1) - \mathbf{S}(k-2))$  is the estimate value of the image features. Let  ${}^x\mathbf{P}_E(k) = \mathbf{U}(k)$ ; then, according to Equation (3), the control law is formed by:

$$\dot{\mathbf{U}}(k) = {}^x\dot{\mathbf{P}}_E(k) = -\lambda \hat{\mathbf{J}}^+(k) \dot{\mathbf{S}}(k) = -\lambda \hat{\mathbf{J}}^+(k) \mathbf{S}^{\oplus}(k) \tag{28}$$

where  $\lambda$  is the control rate, and  $\hat{\mathbf{J}}^+(k)$  is the inverse Jacobian matrix.

**Figure 4.** The scheme of global-state-space IBVS, which is attached to a Robust KF Cooperate with ENN learning. Online weights update to ensure robotic global stability manipulation.



## 5. Simulation and Experimental Results

### 5.1. The State Definition of Robotic System

The simulation data consist of four-feature-points, which are used for robot manipulation. The desired features vector  $\mathbf{S}^*(k)$  does not change over time, and therefore can be calculated before the main control loop of the experiment. On the other hand, the feature vector  $\mathbf{S}(k)$  is not constant due to the camera motion and is obtained in each time instant. The features vector is obtained through:

$$\mathbf{S}(k) = [s_1 \ s_2 \ s_3 \ s_4]^T = [u_1 \ v_1 \ u_2 \ v_2 \ u_3 \ v_3 \ u_4 \ v_4]_{8 \times 1}^T \quad (29)$$

where  $s_i = [u_i \ v_i]^T$ .

Supposing the linear velocity of the end-effector is  $\mathbf{V}(k) = [v_x, v_y, v_z]^T$ , the angular velocity is  $\mathbf{W}(k) = [w_x, w_y, w_z]^T$ , and the robotic control variable is  $\mathbf{U}(k) = [\mathbf{V}(k), \mathbf{W}(k)]_{6 \times 1}^T$ . According to Equation (4), the  $8 \times 6$  Jacobian matrix is given by:

$$\mathbf{J}(k) = \frac{\partial \mathbf{S}(k)}{\partial \mathbf{U}(k)} = \begin{bmatrix} \frac{\partial u_1}{\partial v_x} & \frac{\partial u_1}{\partial v_y} & \frac{\partial u_1}{\partial v_z} & \frac{\partial u_1}{\partial w_x} & \frac{\partial u_1}{\partial w_y} & \frac{\partial u_1}{\partial w_z} \\ \frac{\partial v_1}{\partial v_x} & \frac{\partial v_1}{\partial v_y} & \frac{\partial v_1}{\partial v_z} & \frac{\partial v_1}{\partial w_x} & \frac{\partial v_1}{\partial w_y} & \frac{\partial v_1}{\partial w_z} \\ \frac{\partial u_4}{\partial v_x} & \frac{\partial u_4}{\partial v_y} & \frac{\partial u_4}{\partial v_z} & \frac{\partial u_4}{\partial w_x} & \frac{\partial u_4}{\partial w_y} & \frac{\partial u_4}{\partial w_z} \\ \frac{\partial v_4}{\partial v_x} & \frac{\partial v_4}{\partial v_y} & \frac{\partial v_4}{\partial v_z} & \frac{\partial v_4}{\partial w_x} & \frac{\partial v_4}{\partial w_y} & \frac{\partial v_4}{\partial w_z} \end{bmatrix}_{8 \times 6} \quad (30)$$

According to Equation (5), the size  $48 \times 1$  of the system state vector given by:

$$\mathbf{X}(k) = [j_{11} \ j_{12} \ j_{13} \ j_{14} \ j_{15} \ j_{16} \ \dots \ j_{81} \ j_{82} \ j_{83} \ j_{84} \ j_{85} \ j_{86}]_{48 \times 1}^T \quad (31)$$

where  $j_{ik}$  refers to the  $i$ th row and  $k$ th column of  $\mathbf{J}(k)$ .

## 5.2. Simulation Evaluation

An eye-in-hand simulation environment was set up to test the performance of the proposed method. Camera movement covers linear, rotational movement, as well as the combination of the translational and rotational movements in the workspace. In simulation, we consider only two difficult tasks, which consist of four cases (combined movement and rotation movement). The evaluation goals of Cases 1 and 2 are test the feature trajectories and the camera trajectory performances of PBVS, IBVS, and our method. Cases 3 and 4 test the performance's global stability and robustness for both KBF and our method.

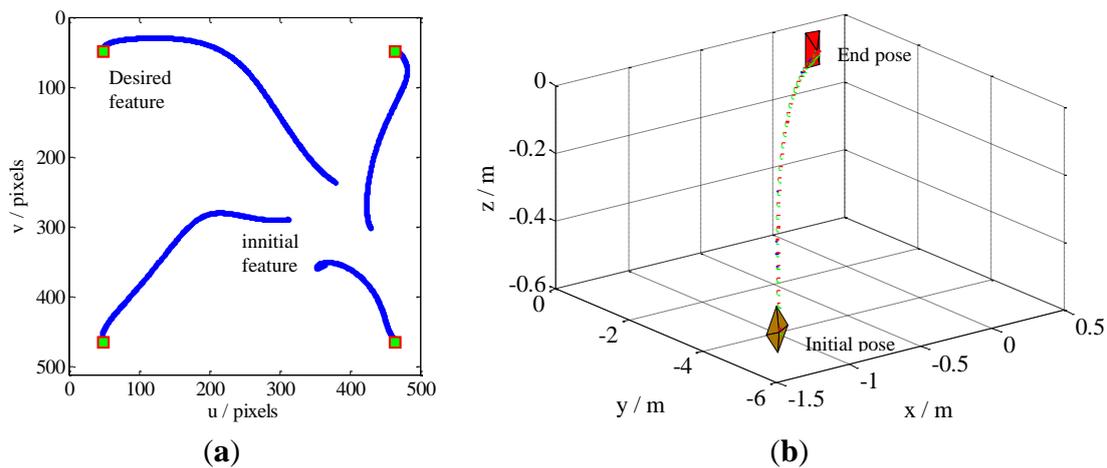
Case 1 involves a combination of the translational and rotational movements of the camera. The results are illustrated in Figure 5a. The feature trajectories on the image plane, obtained through our method, are constrained on the camera field-of-view (FOV), but the PBVS method for this task the feature trajectories easy leaves the FOV (Figure 6a). Additionally, Figure 5b shows that the camera trajectory in the Cartesian space obtained by our method moves on a seemingly straight line from the initial pose toward the desired pose. This result is similar to that obtained by the PBVS method, as illustrated in Figure 6b. The IBVS method for the same task, the feature trajectories are constrained by the controller to move on straight lines from the initial toward desired feature points (Figure 7a), but the camera motion in the Cartesian space (Figure 7b) becomes slight odd, *i.e.*, the camera undergoes an abrupt movement, that is not perfect compared with our method.

Case 2 involves a pure camera rotational movement around the optical axis. The feature points obtained by our method (Figure 8a) move in almost straight lines on the image plane; this movement is very similar to those obtained by the IBVS method. This is because our method is similar to IBVS in that we increase redundant manipulation on the Y-axis to minimize the image feature trajectories

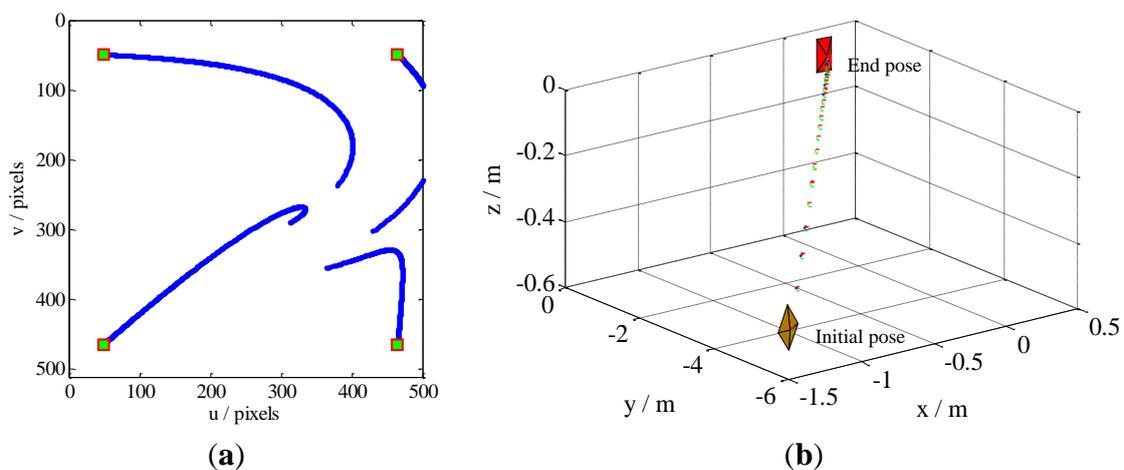
(Figure 8b). Additionally, as illustrated in Figure 8c, the PBVS method converge to the desired pose as the image feature points are constrained by the controller to move on circular trajectories. In such circumstances, the feature points almost leave the FOV.

Therefore, compared with the IBVS and PBVS methods, our approach utilizes the advantages of the PBVS method to improve the camera motion trajectory, and takes advantages of the IBVS method to constrain the feature trajectories to avoid situations where features points leave the FOV.

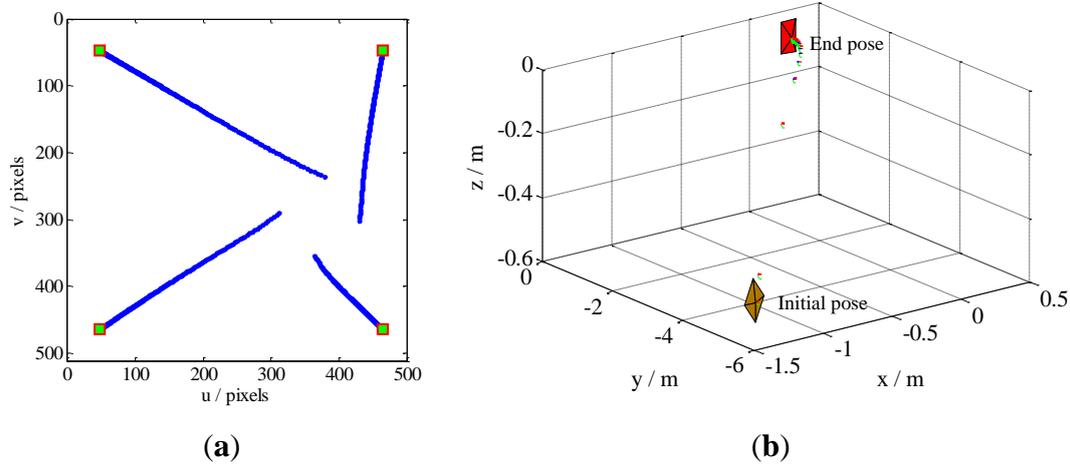
**Figure 5.** Results obtained by our method for Case 1. The sampling interval is 0.1, and the control rate  $\lambda$  is 0.15. Our method does not require camera parameters. The feature trajectories are constrained on the FOV, and the camera trajectory moves in almost a straight line, from the initial pose toward the desired pose. (a) The feature trajectories; (b) The camera trajectory.



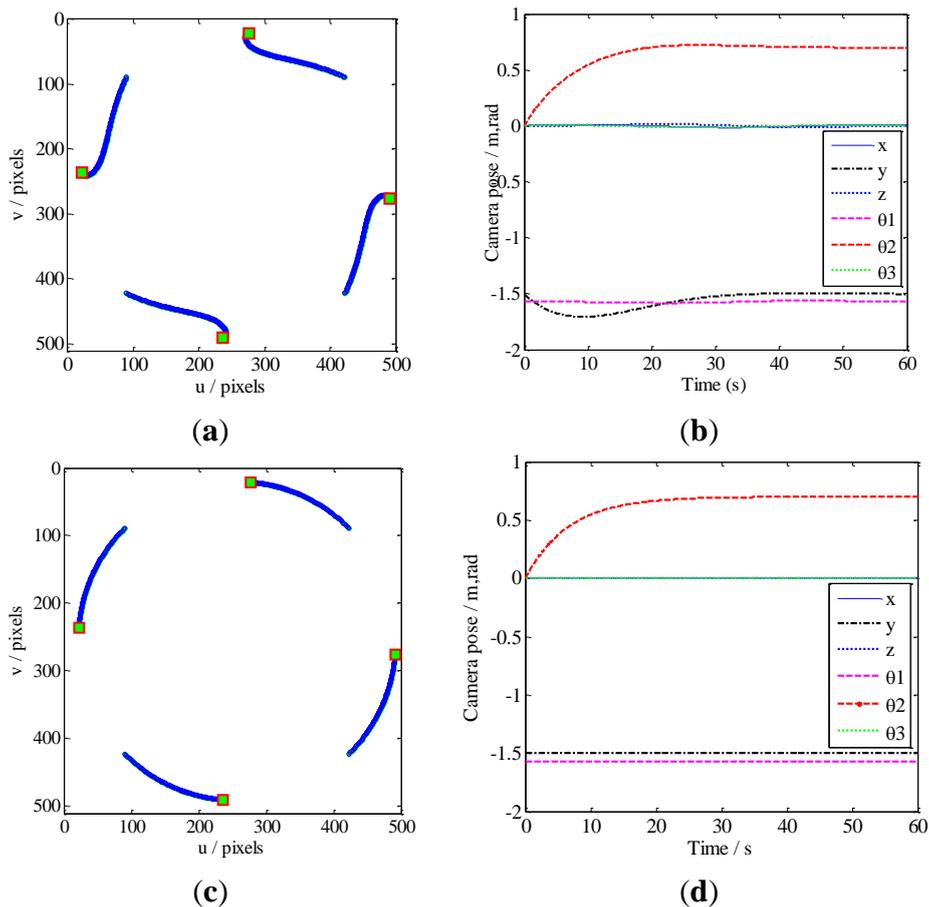
**Figure 6.** Results obtained by the PBVS method for Case 1. The intrinsic camera parameters are chosen as  $u_0 = v_0 = 256$  and  $fk_u = fk_v = 1,000$ . The camera trajectory moves in a seemingly straight line from the initial pose toward desired pose, but the feature trajectories leave the FOV. (a) The feature trajectories; (b) The camera trajectory.



**Figure 7.** Results obtained by the IBVS method for Case 1. The camera parameters are the same as in the PBVS method. The feature trajectories are constrained on the FOV, but the camera trajectory becomes slightly odd, *i.e.*, the camera moves abruptly. **(a)** The feature trajectories; **(b)** The camera trajectory.



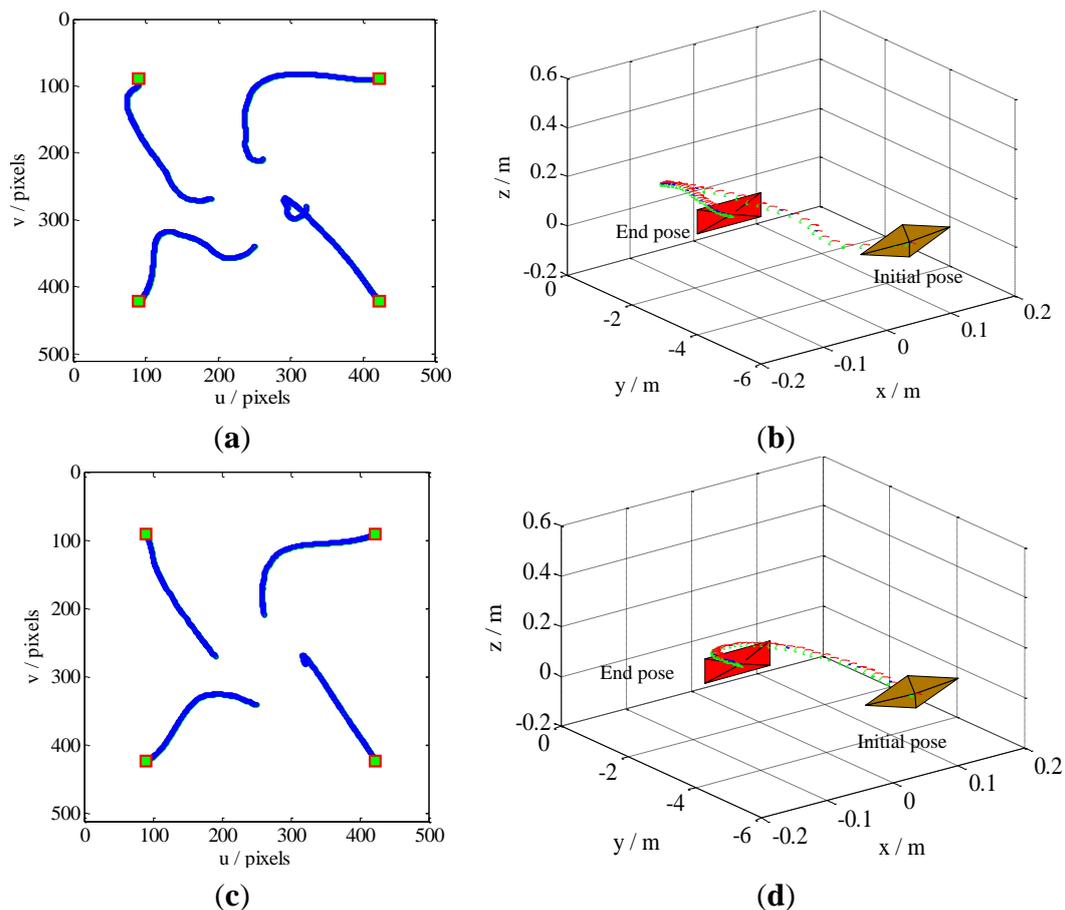
**Figure 8.** Comparison of our method with the PBVS method for Case 2. The feature trajectories are constrained on the FOV by our method, while the results of PBVS almost leave the FOV. **(a)** Feature trajectories by our method; **(b)** The camera pose by our method; **(c)** Feature trajectories by PBVS; **(d)** The camera pose by PBVS.



Furthermore, because for PBVS, planar homography is used for pose estimation of the object with respect to the end-effector, and because IBVS is associated with Jacobian computation, both methods are sensitive to camera calibration error. In works [35], the influence of the calibration parameters on the system is investigated. One solution to this problem uses intelligent hybrid control laws (as proposed in [19]) by introducing neural network reinforcement learning. In our method, the Jacobian is an online estimation, which has nothing to do with camera calibration and system modeling errors. Therefore, our method avoids the corrupted performances caused by calibration and modeling errors.

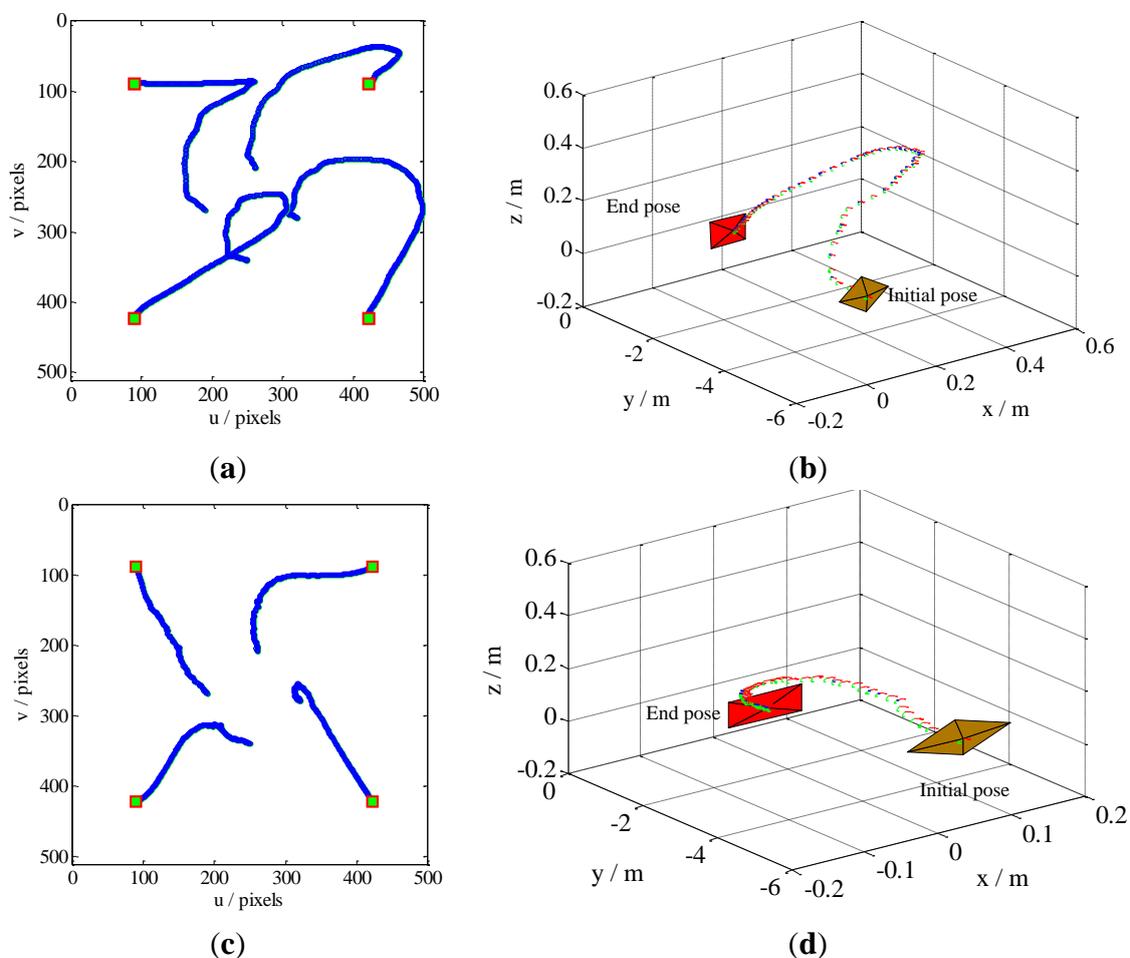
Case 3 deals with dynamic performances in a noisy environment. This case examines the stability of our method, in comparison with the KBF method [21], in a harsh noisy environment. At each sampling instant, uniformly distributed random noises are added to the control system. A Gaussian noise sequence with zero mean, where the variances are  $1 \times 10^{-3}$ , were added to the state and observation models (the additive noises shown in Figure 4), respectively. The results, shown in Figure 9, are compared with KBF's. In the former, the camera movement and feature trajectories are smoother and more preferable.

**Figure 9.** Comparison of our method with KBF for Case 3. The additive noises are set at zero mean, and variances are set at  $1 \times 10^{-3}$ . **(a)** Features trajectories by KBF; **(b)** Camera trajectory by KBF, camera with slight retreat; **(c)** Feature trajectories by our method; **(d)** Camera trajectory by our method, camera without retreat.



When the variance is  $9 \times 10^{-3}$ , the camera pose in the Cartesian space and the feature trajectories on the image plane for this situation are given in Figure 10. It is obvious that the results of the KBF method are seriously deteriorated by the additive noises; the task goals of reaching the desired position and zeroing the errors are accomplished, but the camera engages in unnecessary retreat, and the image features in the image plane almost leave the FOV. Not only does our method enable convergence on the desired pose, it also robustly performs in the presence of noise. On the other hand, when the noise variances are around  $1 \times 10^{-2}$ , The KBF method failure to have end-effector positioning, *i.e.*, the camera randomly moves in Cartesian space and the image features in image plan are unknown. In contrast, our method always converges to the desired pose without camera retreat. It is clear that our method has global stability irrespective of the presence of noise.

**Figure 10.** Comparison of our method with KBF for Case 3. The additive noises are set at zero mean, and the variances are set at  $9 \times 10^{-3}$ . (a) Feature trajectories by KBF; (b) Camera trajectory by KBF, camera with serious retreat; (c) Feature trajectories by our method; (d) Camera trajectories by our method, camera without retreat.



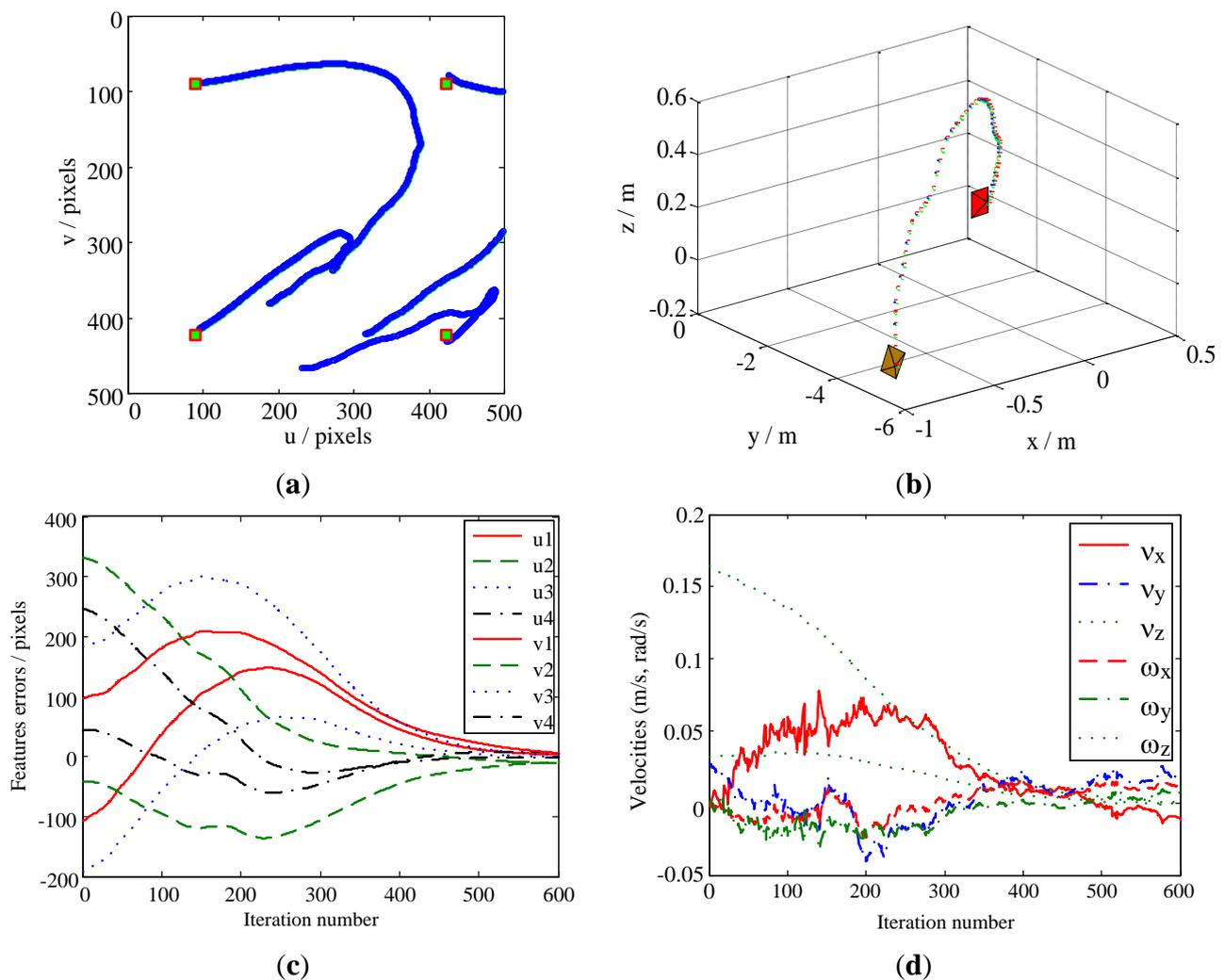
Case 4 deals with dynamic performances due to system destabilization. During the actual robot manipulation, the statistical characteristics of model noise and observation noise are variational, which leads to system destabilization. For simplicity, but without loss of generality, we consider the statistical parameters  $\mathbf{Q}(k)$  and  $\mathbf{R}^*(k)$  (shown in Figure 2) are classical situation, and then a challenging

experiment have been implemented to simulate the robot manipulation, in this paper, a value of  $\mathbf{Q}(k) = 2^*(\mathbf{E})_{48 \times 48}$  and  $\mathbf{R}(k) = 0.04^*(\mathbf{E})_{8 \times 8}$  are chosen for the covariance of noises, where  $\mathbf{E}$  is the unit matrix.

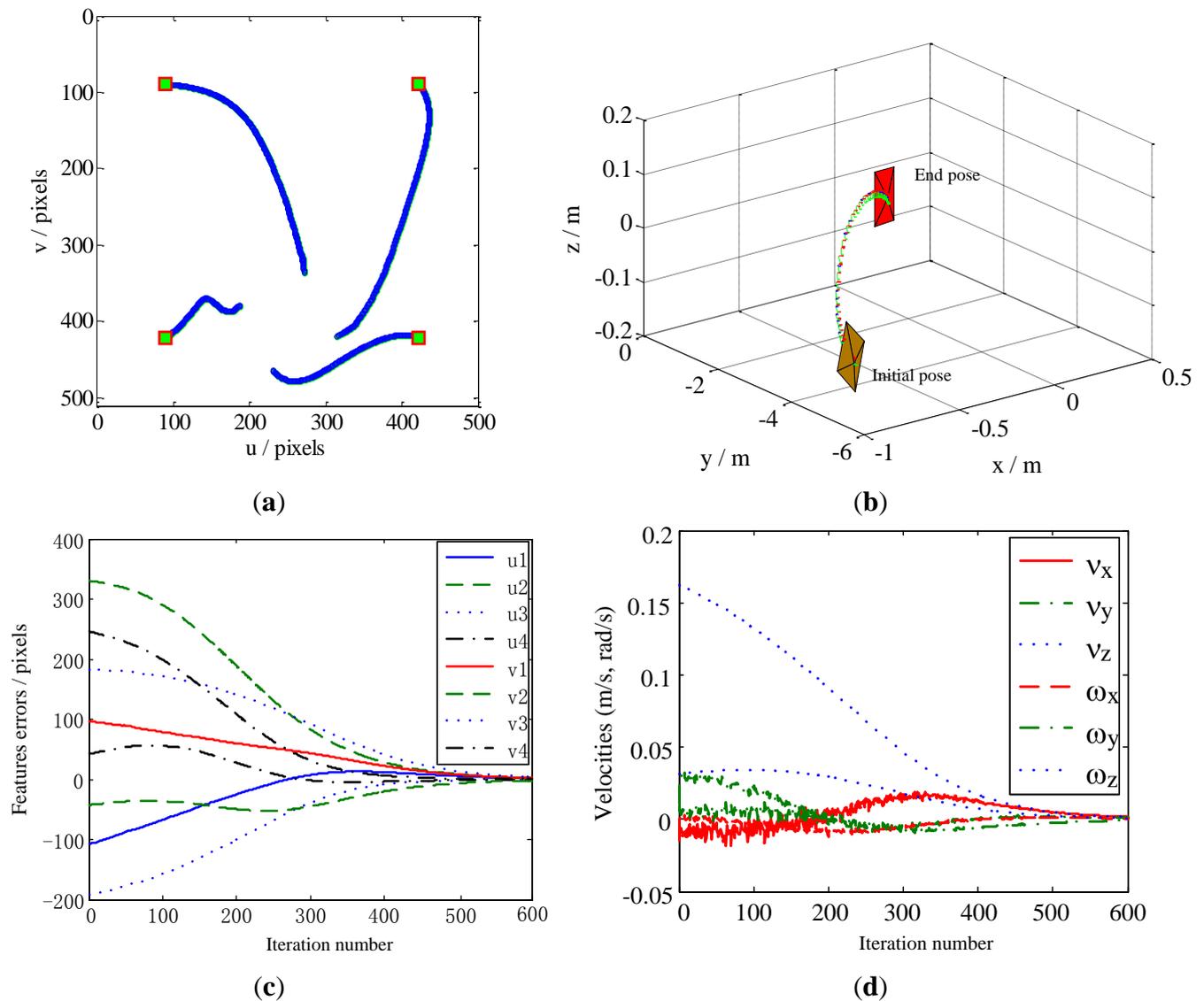
The dynamic performance of the KBF method is shown in Figure 11. Here, the image features leave the FOV (Figure 11a), and the velocity of the camera does not zero (Figure 11d). As a result, the camera pose fails to converge toward the desired pose at the steady state. For the same case, the performance of our method is presented in Figure 12. It can be seen that the local minima problem of the KBF method has been avoided, and the convergence of both the camera pose and the image features toward their desired targets has been achieved.

On the other hand, different values of  $\mathbf{Q}(k)$  and  $\mathbf{R}(k)$  are considered for different simulations. The results tell us that when the covariance of noises changes slightly, the results of the KBF method will alter largely, and even lose their tracking ability (due to serious camera retreat). When the robust KF was selected to cooperate with ENN learning, our method always worked well, even though the  $\mathbf{Q}(k)$  and  $\mathbf{R}(k)$  were changing in a larger region. This means that the proposed global-state-space IBVS is the robust despite system destabilization.

**Figure 11.** Results obtained by the KBF method for Case 4. (a) Feature trajectories; (b) Camera trajectory; (c) Feature errors; (d) Camera velocities.



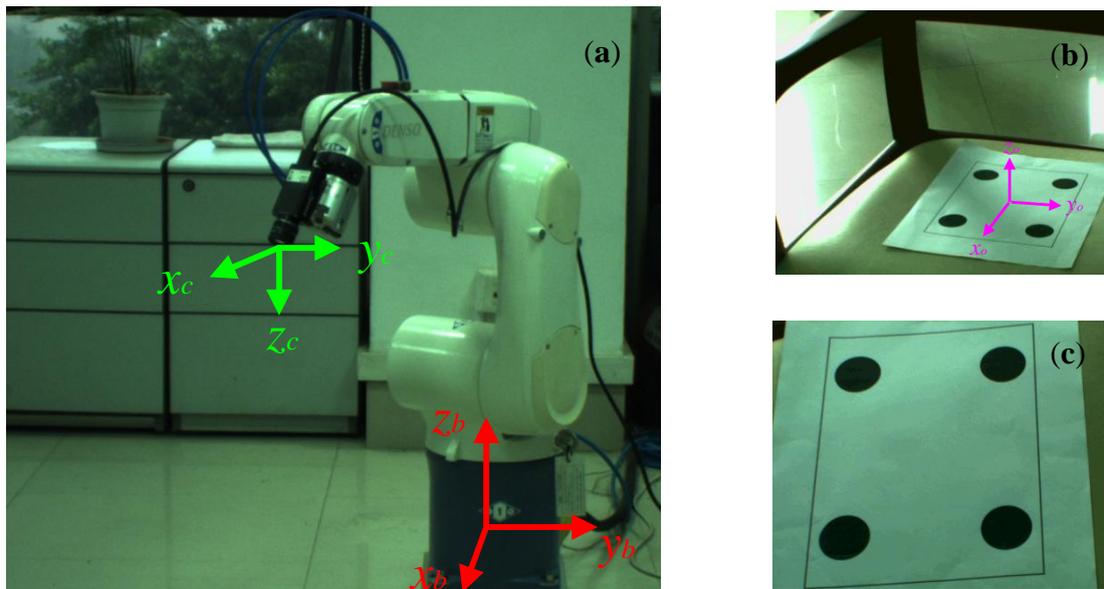
**Figure 12.** Results obtained by our method for Case 4. (a) Feature trajectories; (b) Camera trajectory; (c) Feature errors; (d) Camera velocities.



### 5.3. Experimental Results and Discussion

The experimental results have been carried out using eye-in-hand configurations (Figure 13a). The task is the online dynamic estimation of a Jacobian  $J(k)$  so as to drive the end-effector from the initial pose (Figure 13b) to the desired pose (Figure 13c). Our VS system consists of a *DENSO* RC7M-VSG6BA robotic controller, a computer with an Intel Corei5 2.67-GHz CPU, 4GBs of RAM for image processing (The robotic controller and image processing computer can communicate through RS232C serial interface), and a *DENSO* VS-6556GM six-DoF robotic manipulator with a Basler scA1300-32fm/fc camera mounted at its end-effector. The object is an A4 paper with four black-colored small circular disks on it. The object images are captured by the camera at a rate of 30 Hz. The resolution is  $640 \times 480$ , and the center points of the small circular disks are used as feature points.

**Figure 13.** (a) Experimental environment with eye-in-hand configurations. (b) Initial features. (c) Desired features.



Experiments for the following cases were performed: Case 1 deals with pure rotational movement of the camera, with initial features  $\mathbf{S}(k) = (481.1 \ 158.9, 244.2 \ 135.2, 220.4 \ 372.1, 457.3 \ 395.8)$  and desired features  $\mathbf{S}^*(k) = (319.9 \ 19.3, 154.0 \ 190.1, 324.8 \ 356.0, 490.7 \ 185.2)$ . Case 2 deals with translational movement of the camera, with the initial features  $\mathbf{S}(k) = (217.5 \ 38.1, 89.3 \ 38.1, 89.3 \ 166.3, 217.5 \ 166.3)$  and desired features  $\mathbf{S}^*(k) = (595.3 \ 202.4, 416.7 \ 202.4, 416.7 \ 381.0, 595.3 \ 381.0)$ . Case 3 deals with a combination of translational and rotational movement of the camera, with the initial features  $\mathbf{S}(k) = (225.2 \ 299.0, 103.9 \ 286.8, 91.7 \ 408.1, 213.0 \ 420.3)$  and desired features  $\mathbf{S}^*(k) = (534.8 \ 81.0, 342.9 \ 162.1, 424.0 \ 354.0, 615.9 \ 272.9)$ . The value of  $\lambda = 0.35$  was adopted for the control rate.

**Figure 14.** Experimental results obtained by our method for Case 1 rotational movement. (a) Feature point trajectories; (b) Camera trajectory; (c) Feature errors; (d) Camera velocities.

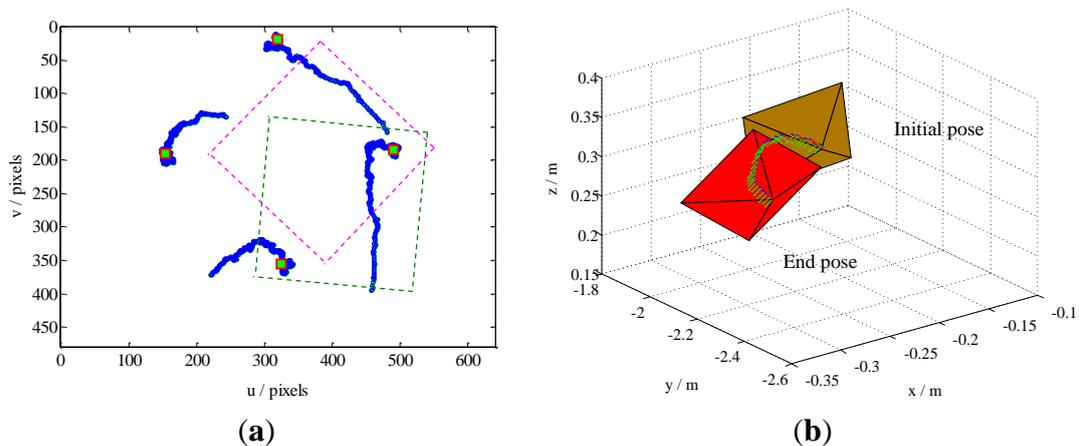
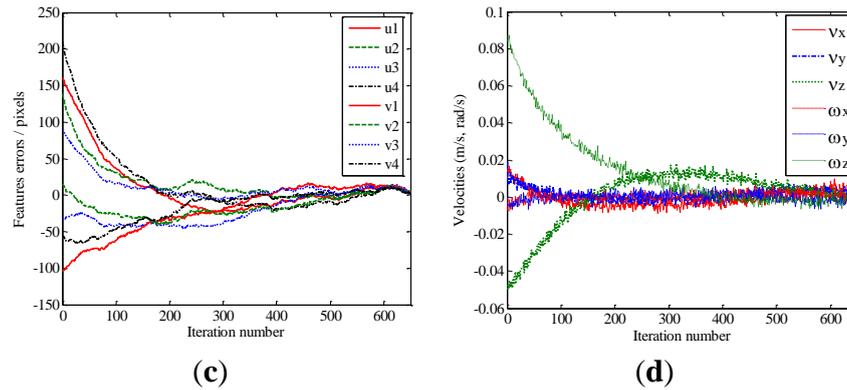
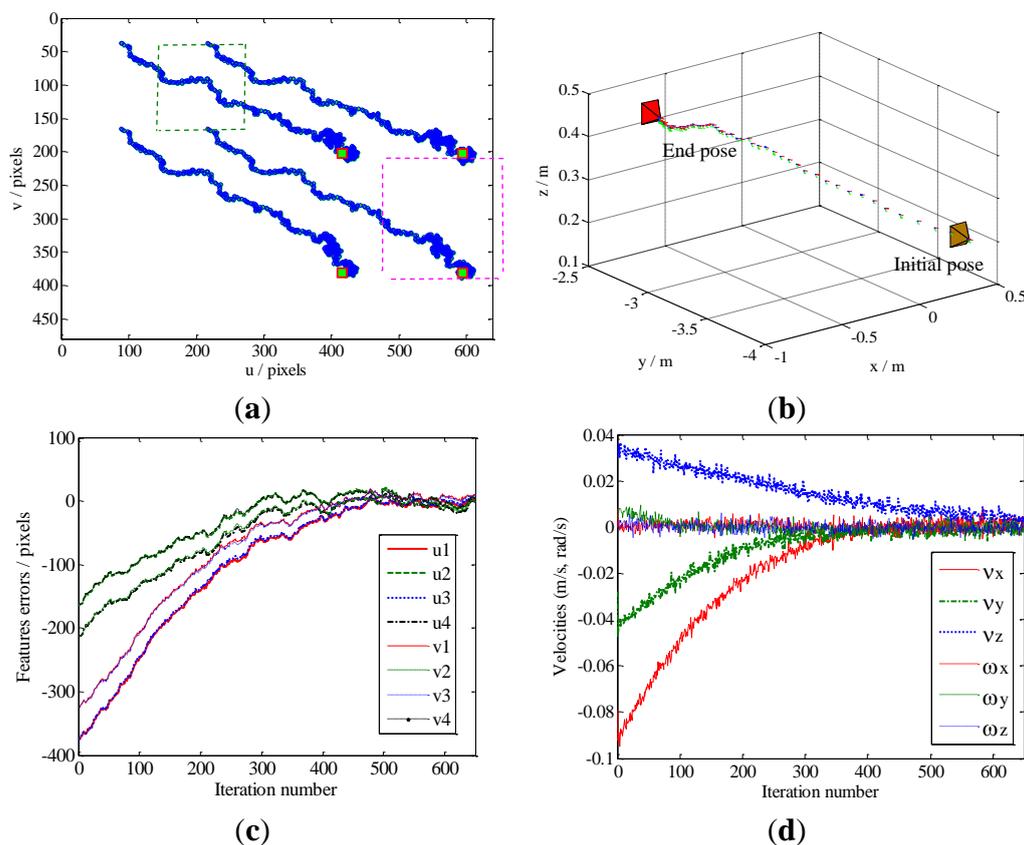


Figure 14. Cont.

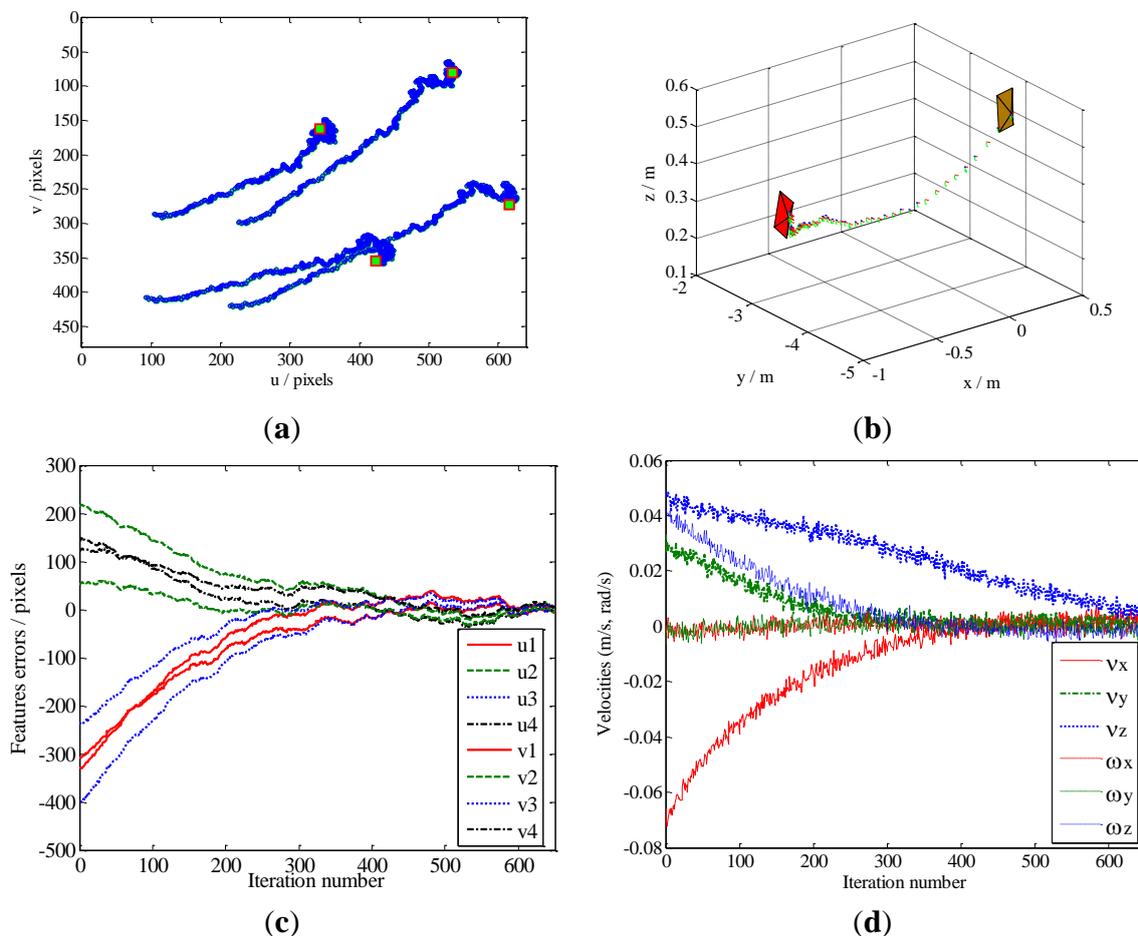


The experimental results are shown in Figures 14–16. For Case 1 and Case 3, the camera involves rotational movement with low rotational angle values (due to the limitation of the joints angle and the workspace). The results of the image error are minimized, with the initial features converging toward the desired features. The feature trajectories are almost straight lines on the image plane, and the steady-state errors in the image space are in order of 4 pixels for Case 1 and Case 3, while the errors are less than 6 pixels for Case 2. In addition, the camera trajectory in the Cartesian space obtained by our method show that all the tasks of the three cases are completed with global stability without camera retreat. Moreover, the image features remain within the FOV.

**Figure 15.** Experimental results obtained by our method for Case 2 translational movement. (a) Feature point trajectories; (b) Camera trajectory; (c) Feature errors; (d) Camera velocities.



**Figure 16.** Experimental results obtained by our method for Case 3 translational and rotational movement. **(a)** Feature point trajectories; **(b)** Camera trajectory; **(c)** Feature errors; **(d)** Camera velocities.



In the experiment, our approach direct control of the feature points on the image plane for robotic task manipulation, in other words, the proposed method of observing the changing of features on image plane direct control to the robot converging toward the desired pose with six-degree-of-freedom. As described in Section 2, if the feature errors converge toward zero that meaning is the robot successful achieve the task manipulation. Therefore, the experimental results can validate the proposed method.

## 6. Conclusions

In this work, a new global-state-space IBVS scheme for uncalibrated model-independent robot manipulation in an eye-in-hand configuration is discussed. Here, a robust KF cooperates with ENN learning techniques so as to ensure robust stability in global-state-space with respect to image features within the camera field-of-view (FOV). Also, the image Jacobian matrix is estimated without requiring camera parameters and depth information. Therefore, our method avoids the corrupted performances caused by calibration and modeling errors. Through various simulation and experimental results through IBVS, PBVS, KBF and our method, we have shown that our approach takes the advantages of PBVS to improve the camera moving trajectory, and takes the advantages of IBVS to avoid the loss of

image features. Finally, in comparison with the KBF method, our method is robust despite outside and system noises.

## Acknowledgements

This work was supported in part by the Fujian Provincial Natural Science Foundation of China (NO.2010J05141).

## Conflicts of Interest

The authors declare no conflict of interest.

## References

1. Shirai, Y.; Inoue, H. Guiding a robot by visual feedback in assembling tasks. *Pattern Recognit.* **1973**, *5*, 99–108.
2. Chaumette, F.; Hutchinson, S. Visual servo control. Part II: Advanced approaches. *IEEE Robot. Autom. Mag.* **2007**, *14*, 109–118.
3. Jorge, P.; Pablo, G.; Fernando, T. Visual control of robots using range images. *Sensors* **2010**, *10*, 7303–7322.
4. Garcia-Aracil, N.; Perez-Vidal, C.; Sabater, J.M.; Morales, R.; Badesa, F.J. Robust and cooperative image-based visual servoing system using a redundant architecture. *Sensors* **2011**, *11*, 11885–11900.
5. Do-Hwan, P.; Jeong-Hoon, K.; In-Joong, H. Novel position-based visual servoing approach to robust global stability under field-of-view constraint. *IEEE Trans. Indus. Elec.* **2012**, *59*, 4735–4752.
6. Chaumette, F.; Hutchinson, S. Visual servo control. Part I: Basic approaches. *IEEE Robot. Autom. Mag.* **2006**, *13*, 82–90.
7. Hutchinson, S.; Hager, G.; Corke, P. A tutorial on visual servo control. *IEEE Trans. Robot. Autom.* **1996**, *12*, 651–670.
8. Baumann, M.; Leonard, S.; Croft, E.A.; Little, J.J. Path planning for improved visibility using a probabilistic road map. *IEEE Trans. Robot.* **2010**, *26*, 195–200.
9. Chaumette, F. Potential Problems of Stability and Convergence in Image-Based and Position Based Visual Servoing. In *The Confluence of Vision and Control, LNCIS Series*; Kriegman, D., Hager, G., Morse, A., Eds.; Springer-Verlag: New York, NY, USA, 1998; Volume 237, pp. 66–78.
10. Garcia-Aracil, N.; Malis, E.; Aracil-Santonja, R.; Perez-Vidal, C. Continuous visual servoing despite the changes of visibility in image features. *IEEE Trans. Robot.* **2005**, *21*, 1214–1220.
11. Chesi, G.; Hashimoto, K.; Prattichizzo, D.; Vicino, A. Keeping features in the field of view in eye-in-hand visual servoing: A switching approach. *IEEE Trans. Robot.* **2004**, *20*, 908–913.
12. Comport, A.; Pressigout, M.; Marchand, E.; Chaumette, F. A Visual Servoing Control Law that is Robust to Image Outliers. In Proceedings of IEEE International Conference on Intelligent Robots and Systems (IROS'03), Las Vegas, NV, USA, 27–31 October 2003; Volume 1, pp. 492–497.

13. Sanderson, A.C.; Weis, L.E.; Neuman, C.P. Dynamic sensor-based control of robots with visual feedback control. *IEEE J. Robot. Autom.* **1987**, *7*, 31–47.
14. Redwan, D.; Nicolas, A.; Youcef, M.; Omar, A.-A.; Philippe, M. Dynamic visual servoing from sequential regions of interest acquisition. *Int. J. Robot. Res.* **2012**, *4*, 1–19.
15. Youcef, M.; Francois, C. Path planning for robust image-based control. *IEEE Trans. Robot. Autom.* **2002**, *18*, 534–549.
16. Chesi, G. Visual servoing path-planning via homogeneous forms and LMI optimizations. *IEEE Trans. Robot. Autom.* **2009**, *25*, 281–291.
17. Chesi, G.; Shen, T. Conferring robustness to path-planning for image-based control. *IEEE Trans. Control Syst. Tech.* **2011**, doi: 10.1109/TCST.2011.2157346.
18. Eissa, N.; Aleksandar, V.; Janabi-Sharifi, F. A second-order conic optimization-based method for visual servoing. *Mechatronics* **2012**, *22*, 444–467.
19. Zoran, M.; Marko, M.; Mihailo, L.; Bojan, B. Neural network reinforcement learning for visual control of robot manipulators. *Exp. Syst. Appl.* **2013**, *40*, 1721–1736.
20. Shahamiri, M.; Jagersand, M. Uncalibrated Visual Servoing Using a Biased Newton Method for On-Line Singularity Detection and Avoidance. In Proceedings of 2005 IEEE/RSJ International Conference on Intelligent Robots and Systems, 2005 (IROS 2005), Edmonton, AB, Canada, 2–6 August 2005; pp. 3953–3958.
21. Jagersand, M.; Fuentes, O.; Nelson, R. Experimental Evaluation of Uncalibrated Visual Servoing for Precision Manipulation. In Proceedings of 1997 IEEE International Conference on Robotics and Automation, Albuquerque, NM, USA, 20–25 April 1997, pp. 2874–2880.
22. Piepmeier, J.A.; Lipkin, H. Uncalibrated eye-in-hand visual servoing. *Int. J. Robot. Res.* **2003**, *22*, 805–819.
23. Azad, S.; Amir-Massoud, F.; Martin, J. Robust Jacobian Estimation for Uncalibrated Visual Servoing. In Proceedings of 2010 IEEE International Conference on Robotics and Automation (ICRA), Anchorage, AK, USA, 3–7 May 2010; pp. 5564–5569.
24. Qian, J.; Su, J. Online estimation of image Jacobian matrix by Kalman-Bucy filter for uncalibrated stereo vision feedback. In Proceedings of IEEE International Conference on Robotics and Automation, 2002 (ICRA '02), Washington, DC, USA, 14–18 May 2002; PP. 562–567.
25. Prem Kumar, P.; Laxmidhar, B. Visual servoing of redundant manipulator with Jacobian matrix estimation using self-organizing map. *Robot. Autom. Syst.* **2010**, *58*, 978–990.
26. Kosmopoulos, D.I. Robust Jacobian matrix estimation for image-based visual servoing. *Robot. Comp. Integr. Manufac.* **2011**, *27*, 82–87.
27. Swagat, K.; Laxmidhar, B.; McGinnity, T.M. Kinematic control of a redundant manipulator using an inverse-forward adaptive scheme with a KSOM based hint generator. *Robot. Autom. Syst.* **2010**, *58*, 622–633.
28. Sun, G.; Scassellati, B. A fast and efficient model for learning to reach. *Int. J. Hum. Robot.* **2005**, *4*, 391–413.
29. Chui, C.K.; Chen, G. *Kalman Filtering with Realtime Application*; Springer-Verlag: Berlin/Heidelberg, Germany, 1987.
30. Simo, S. On unscented kalman filtering for state estimation of continuous-time nonlinear systems. *IEEE Trans. Auto. Control* **2007**, *52*, 1631–1641.

31. Elman, J.L. Finding structure in time. *Cognit. Sci.* **1990**, *14*, 179–211.
32. Shi, X.H.; Liang, Y.C.; Lee, H.P.; Lin, W.Z.; Xu, X.; Lim, S.P. Improved Elman networks and applications for controlling ultrasonic motors. *Appl. Artif. Intell.* **2004**, *18*, 603–629.
33. Assal, S.F.M.; Watanabe, K.; Izumi, K. Neural network-based kinematic inversion of industrial redundant robots using cooperative fuzzy hint for the joint limits avoidance. *IEEE Trans. Mech.* **2006**, *11*, 593–603.
34. Martin, P.; Millan, J.R. Robot arm reaching through neural inversions and reinforcement learning. *Robot. Autonom. Syst.* **2000**, *31*, 227–246.
35. Ceren, Z.; Altug, E. Image based and hybrid visual servo control of an unmanned aerial vehicle. *J. Intell. Robot. Syst.* **2012**, *65*, 325–344.

© 2013 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 license (<http://creativecommons.org/licenses/by/3.0/>).