Hybrid regression model via multivariate adaptive regression spline and online sequential extreme learning machine and its application in vision servo system

Zhiyu Zhou¹, Jiangfei Ji¹, Yaming Wang², Zefei Zhu³ and Ji Chen¹

Abstract
To solve the problems of slow convergence speed, poor robustness, and complex calculation of image Jacobian matrix in image-based visual servo system, a hybrid regression model based on multiple adaptive regression spline and online sequential extreme learning machine is proposed to predict the product of pseudo inverse of image Jacobian matrix and image feature error and online sequential extreme learning machine is proposed to predict the product of pseudo inverse of image Jacobian matrix and image feature error. In MOS-ELM, MARS is used to evaluate the importance of input features and select specific features as the input features of online sequential extreme learning machine, so as to obtain better generalization performance and increase the stability of regression model. Finally, the method is applied to the speed predictive control of the manipulator end effector controlled by image-based visual servo and the prediction of machine learning data sets. Experimental results show that the algorithm has high prediction accuracy on machine learning data sets and good control performance in image-based visual servo.

Keywords
Multivariate adaptive regression spline, online sequential extreme learning machine, hybrid regression model, vision servo, robot manipulator

Date received: 6 September 2021; accepted: 3 June 2022

Introduction
The robotic system is applied more and more widely. Visual servo control is an important control measure in a robotic system. Based on visual feedback, visual servo can be classified as position-based (PBVS), image-based (IBVS) and hybrid-based visual servo (HBVS). The PBVS control system needs to use the image information collected by the camera for three-dimensional (3D) reconstruction to obtain the current pose. The hybrid visual servo control obtains the objective image information from the camera and calculates the homography matrix by extracting image features.

In the IBVS, the image Jacobi matrix is an important concept, which represents the relationship between the...
pose change of the robot end effector and the image feature change. In this case, the control relationship can be obtained by the inverse of the matrix. Furthermore, by controlling the robot limb to move to the desired pose, we can obtain the desired features that converge with the image features. As mentioned above, because the inverse of a matrix is required to obtain the control relationship, the singularity problem needs to be addressed.\(^\text{12}\) IBVS only uses image features as the control signal of the system objective function and does not need 3D reconstruction, so it has the characteristics of simple control structure. In addition, the acquisition of control signals in the IBVS system does not depend on the system parameters of the camera or the robot, so IBVS is more suitable for the case where the visual servo is not calibrated. Even with a 2D image, for the uncalibrated IBVS, the information from the image that can be used as a control signal is still significant. The image characteristics of familiar visual servo control include point features, line features,\(^\text{13}\) and image moments,\(^\text{14−16}\) among others. Because the IBVS method is performed in image space, the path of the image features can be controlled better. For example, the feature points can converge to their desired positions in a nearly straight path, which ensures that almost all image features are within the camera lens range. In the study by Liu et al.,\(^\text{17}\) when the unknown geometric parameters for the depth independent Jacobian matrix is taken as the linearization description based on image features, a new error function was established in that study to estimate the parameters.

Through the analysis of IBVS, it can be found that the implementation of IBVS control system is relatively simple and does not need complex 3D reconstruction. In IBVS, the acquisition of image Jacobian matrix is complex. Image Jacobian matrix needs good camera calibration and feature depth information. However, in practice, the depth information of features is not easy to obtain, and the camera calibration is easily disturbed by external noise, which leads to a large difference between the obtained image Jacobian matrix and the actual Jacobian matrix. Therefore, some scholars proposed to use neural network in IBVS. Kang et al.\(^\text{18}\) proposed an adaptive visual servo control method combining extreme learning machine (ELM) and Q-learning. ELM was used to predict Jacobian matrix and pseudo inverse, which solved the possible difficulties in obtaining Jacobian matrix. In addition, the author used Q-learning to improve the convergence of IBVS system. Qiu and Wu\(^\text{19}\) used adaptive neural network in IBVS to complete the servo task without depth information.

Neural networks can be used to approximate nonlinear parameters. However, traditional neural networks have problems of low approximation accuracy and poor generalization performance. Online sequential extreme learning machine (OS-ELM) is an improved method of neural network ELM. OS-ELM continuously adjusts the output weight of the network through data samples to improve the generalization of the network and the prediction accuracy of the network. Therefore, compared with traditional neural networks, OS-ELM can be used in scenarios that require higher prediction accuracy. Gao et al.\(^\text{20}\) used an online sequential parallel extreme learning machine to predict the duration of use of electronic equipment, and their experiments showed that the OS-ELM method has better stability and higher prediction accuracy. Peng et al.\(^\text{21}\) proposed an adaptive OS-ELM for predicting the power fluctuation of ships. The author verified in the experimental part that the proposed method can predict better power data and improve the performance of the control system. To avoid overfitting when dealing with large-scale data, a method of regularizing OS-ELM is used in data prediction,\(^\text{22}\) and the experimental results show the superiority of OS-ELM.

Considering the complexity of image Jacobian matrix and image feature data in IBVS, MARS is used to select features, reduce the complexity of data, and select better inputs for OS-ELM model, so as to establish a regression model for predicting the product of pseudo inverse of Jacobian matrix and image feature error. MARS can be used for feature selection to improve the accuracy of pattern recognition. Kao and Chiu\(^\text{23}\) applied MARS and recurrent neural network to the problem of variable selection, which improved the performance of engineering control. Li et al.\(^\text{24}\) proposed a new strategy using cluster analysis. MARS is used to describe complex nonlinear relationship models, which improves the accuracy of prediction. Depren et al.\(^\text{25}\) used MARS as an important predictor to determine the amount of nonperforming loans and selected the most suitable parameters from multidimensional data to solve the problem of nonperforming loans in Turkey. Bose et al.\(^\text{26}\) proposed a hybrid model combining MARS and deep neural network (DNN) for stock price prediction, used MARS to screen important data, and passed the screened data to DNN for training. Cartocci et al.\(^\text{27}\) proposed a nonlinear model to characterize the nonlinear redundancy relationship between system signals. MARS is used to identify the relevant fault characteristic matrix in the data, which improves the performance of the diagnosis system.

The IBVS manipulator is a real-time control system that primarily includes objective image acquisition, image processing, objective features extraction, image Jacobian matrix estimation, feedback control sequence calculation, and motion control. In IBVS, the calculation of image Jacobian matrix needs depth information and is easily disturbed by noise in the calculation process. The estimation of image Jacobian matrix can be solved by regression model. Therefore, this article proposes a hybrid regression model based on multivariate adaptive regression spline and OS-ELM. Using the good prediction ability of OS-ELM, the regression model of image Jacobian matrix is established. To improve the prediction ability of the model as much as possible, the model input should consider as many variables as possible, including image feature information, Jacobian matrix, and the change of manipulator joint angle. At the same time, considering the complexity of model input data, MARS is used to obtain
important input variables and reduce the dimension of OS-ELM model input variables, so as to establish a more efficient multiple adaptive regression spline and online sequential extreme learning machine (MOS-ELM) model. To test the robustness of the hybrid regression model, we first apply it to the machine learning data set and then apply it to the IBVS system simulation model.

(1) Evaluation of the importance of training samples by the multivariate adaptive regression spline (MARS) method, and its selection of important variables samples as the input features for the OS-ELM, which reduces the feature dimension to a certain degree, thus simplifying the computational complexity of the system. The prediction results of machine learning data set show that this method improves the prediction accuracy of regression model.

(2) Our proposed method addresses the difficulty of calculating the precise image Jacobian matrix in the Classical IBVS system and improves the system convergence speed. The performance analysis result shows that the proposed algorithm performs well in the IBVS system with respect to the performance indexes of convergence speed, trajectory length, and cumulative error.

(3) The proposed method has good performance in IBVS systems with different configurations, and MOS-ELM is used to predict the product of image Jacobian matrix and image feature error, which not only solves the difficulty of accurately calculating image Jacobian matrix but also shows good performance in the interference of external noise and improves the robustness of IBVS system.

The structure of this article is as follows: The first section introduces the research status of IBVS and MOS-ELM. The second section introduces MARS and OS-ELM. The model of MOS-ELM is introduced in the third section. Experiments are given in the fourth section. The conclusion is given in the final section.

**Brief review of multivariate adaptive regression spline and OS-ELM**

**Multivariate adaptive regression spline**

MARS is a nonlinear and nonparametric regression model that predicts continuous dependent variables based on a set of independent variables. MARS uses a combination of basis functions to fit complex nonlinear functions and effectively segments high-dimensional data in the case of large samples. In addition, MARS can identify the important variables from among all the variables of the system and reduce the dimensionality of these variables. Thus, it can be used for feature selection. For a training set with input variable \( x = (x_1, \ldots, x_n) \) and corresponding output variable \( y \), the training set containing \( L \) samples can be expressed as \( \{y_i; x_i_1, \ldots, x_i_m\}_{i=1}^{L} \), \( y \) can be approximated by the following linear function \( \hat{f}(x) \):

\[
\hat{y} = \hat{f}(x) = P_0 + \sum_{i=1}^{N} P_i Q_i(x)
\]

where \( \hat{y} \) represents the predicted value of the output variable, \( Q_i(x) \) is the basis function, \( P_i \) is the weight coefficient corresponding to the basis function \( Q_i(x) \), \( N \) is the number of basis functions, and \( P_i \) is defined according to the least square method as

\[
\{P_i\}_0^N = \min_{\{P_i\}_0^N} \sum_{j=1}^{L} \left[ y_j - \sum_{i=0}^{N} P_i Q_i(x) \right]^2
\]

Defined \( Q_i(x) \) as

\[
Q_{2i-1}(x) = (t - x)_+ = \begin{cases} t - x, & t > x \\ 0, & \text{Otherwise} \end{cases}
\]

\[
Q_{2i}(x) = (x - t)_+ = \begin{cases} x - t, & x > t \\ 0, & \text{Otherwise} \end{cases}
\]

where \( t \) represents the node, \( t \in \{x_1, \ldots, x_m\} \), \( j, \ldots, L \), the sign “+” indicates the positive number of a number. MARS algorithm is divided into two processes: forward process and backward pruning process. The backward pruning process results in the optimal model by removing the invalid basis function according to the generalized cross-validation rule ensuring the accuracy of the model.

**Step 1 (forward process):** The forward process is to continuously add new basis functions on the basis that the basis function combination only contains constant terms, so as to find the basis function combination that minimizes model \( \sum_{i=1}^{n} (y_i - \hat{f}(x_i)) \). When the number of generated basis functions reaches the defined number \( N \) or the model accuracy reaches the prediction threshold, the iteration stops.

**Step 2 (backward pruning process)**: In the forward process, as the basis functions increase, the model becomes more and more complex, and there is a risk of overfitting. The backward pruning process is based on the generalized cross validation (GCV) criterion to reduce the basis function and thereby to reduce the risk of overfitting of the regression model. Where GCV is defined as follows

\[
\text{GCV}(N) = \frac{\sum_{i=1}^{L} (y_i - \hat{f}(x))^2}{L \left(1 - \frac{C(N)}{L}\right)^2}
\]

where \( C(N) = dN + N + 1 \), and \( L \) is the samples number, \( d \) is the penalty parameter, which is usually between 2 and 4. \( N \) is the number of basis functions. Finally, we take the sub-model with the smallest GCV value as the final MARS model.
Online sequential extreme learning machine

ELM has been used widely since it was first proposed in 2006. OS-ELM is a variant of the ELM, which is an incremental learning algorithm. Sequential learning algorithms structure a model whose construction is alterable, which is used in dynamic time-varying systems. Its learning process consists of initialization and online learning. In the initialization stage, the output weight matrix \( \beta^{(0)} \) and matrix \( H_0 \) of the OS-ELM are initialized using a small number of samples. From the study of Huang et al., we know that the sample number used in the initial matrix \( H_0 \) should be greater than or equal to the hidden layer node number in the network. Then, during online learning, a single sample or sample data block is used to update the output weight matrix \( \beta^{(0)} \) of the initial stage.

The specific training method is given as follows.

**Step 1** (initialization process): Derive a subset \( D_0 = \{ y_i; x_{1i}, \ldots, x_{ni} \} \) from the sample \( D = \{ y_i; x_{1i}, \ldots, x_{mi} \} \), \( i = 1, \ldots, L \) to be the initial training sample, where \( L_0 \geq M \), and \( M \) is the hidden layer node number. The input weight \( a_i \) and hidden layer threshold \( b_i \) are generated.

**Algorithm 1. MOS-ELM algorithm.**

| Input: Training data \( D = \{ y_i; x_{1i}, \ldots, x_{ni} \} \), activation function \( g(x) \), hidden layer node number \( M \), maximum basis function number \( N \); |
| MARS forward iteration: |
| 1. initialization \( Q_0(x) = 1 \); |
| 2. In each iteration, two new basis functions are generated based on the previous basis functions \( (Q_{2k-1}, Q_{2k}) \); |
| 3. Calculate the Mean Square Error: \( \sum_{i=1}^{L_0} (y_i - \hat{f}(x_i))^2 \); |
| 4. Repeat the previous steps until the number of maximum basis functions for the model attains \( N \); |
| MARS backward pruning: |
| 5. Use the Generalized Cross Validation (GCV) criterion for the basis function deletion: |
| \[ GCV(N) = \frac{\sum_{i=1}^{L_0} (y_i - \hat{f}(x_i))^2}{L(1 - \frac{C(N)}{L})^2} \] |
| 6. MARS output: Feature subset \( D' \) of the sample is selected according to the acquired MARS model; |
| OS-ELM initialization: |
| 7. Select \( L_0 \) samples from \( D' \) as initialization training samples; |
| 8. Randomly initialize input weight \( a_i \) and bias \( b_i \), where \( i = 1, \ldots, L_0 \); |
| 9. Calculate the initial hidden layer output matrix \( H_0 \) as: |
| \[ H_0 = \begin{bmatrix} G(a_1, b_1, x_1) & \cdots & G(a_L, b_M, x_1) \\ \vdots & \ddots & \vdots \\ G(a_1, b_1, x_{L_0}) & \cdots & G(a_L, b_M, x_{L_0}) \end{bmatrix}_{L_0 \times M} \] |
| 10. Initial estimation of output weight matrix is given by: |
| \[ \beta^{(0)} = (H_0^T H_0)^{-1} H_0^T Y_0 \] |
| \[ Y_0 = [y_1^T, y_2^T, \ldots, y_{L_0}^T]^T \] |
| OS-ELM sequence learning: |
| 11. Suppose the number of a batch sample is \( L_i \), then calculate output matrix \( H_i \); |
| 12. Finally, calculate the output weight \( \beta^{(k+1)} \): |
| \[ \beta^{(k+1)} = \beta^{(k+1)} + R_{k+1} \beta^{(k)} \] |
| Output: Selected important features, OS-ELM model. |
randomly, where the activation function is a sigmoid function, \( a_i \) and \( b_i \) in the hidden layer nodes of radial basis function (RBF) are the core and influence factor, respectively. The output matrix \( H_0 \) for calculating the initial hidden layer is given as follows

\[
H_0 = \begin{bmatrix}
G(a_1, b_1, x_1) & \cdots & G(a_L, b_M, x_1) \\
\vdots & \ddots & \vdots \\
G(a_1, b_1, x_n) & \cdots & G(a_L, b_M, x_n)
\end{bmatrix}_{L \times M}
\]  
(5)

Then, the initial estimation of the output weight matrix is attained as

\[
\hat{\beta}(0) = (H_0^T H_0)^{-1} H_0^T T_0
\]  
(6)

where \( T_0 = [t_1^T, t_2^T, \ldots, t_L^T]^T \), set \( k = 0 \).

**Step 2** (sequence learning process): When some new samples are inserted in the model, and, say, the number of these samples is \( L_1 \), then we obtain the following formula based on the concept of ELM

\[
\hat{\beta}(1) = (H_1^T H_1)^{-1} \begin{bmatrix} H_0^T & H_1^T \end{bmatrix} \begin{bmatrix} T_0 \\ T_1 \end{bmatrix}
\]  
(7)

Let \( \hat{\beta}(1) \) be represented by \( \hat{\beta}(0) \), then

\[
\hat{\beta}(1) = \hat{\beta}(0) + (H_1^T H_1)^{-1} H_1^T (T_1 - H_1 \hat{\beta}(0))
\]  
(8)

Thus, we can obtain the following recursive formulæ

\[
P_{k+1} = P_k - P_k H_{k+1}^T (I + H_{k+1} P_k H_{k+1}^T)^{-1} H_{k+1} P_k
\]

\[
\hat{\beta}(k+1) = \hat{\beta}(k) + P_{k+1} H_{k+1}^T (T_{k+1} - H_{k+1} \hat{\beta}(k))
\]  
(9)

Finally, set \( k = k + 1 \), then repeat step 2.

**Proposed method**

**Hybrid regression model based on multivariate adaptive regression spline and OS-ELM**

During the development of OS-ELM, many studies proposed improved algorithms for OS-ELM, such as KOS-ELM, among others. However, the Kalman and online sequential extreme learning machine (KOS-ELM) algorithm suffers from some problems related to training time and accuracy. To address the question of regression, in this study, we proposed the regression model of MARS and OS-ELM. First, we developed a prediction model that uses MARS and takes the training sample as input. One of the characteristics of MARS is the importance of assessment features to the prediction result. Therefore, in our study, we separate the important features from all the features of the sample using this characteristic. Then, these important features are used as input of the OS-ELM. The benefit of eliminating these unnecessary features is to reduce the dimensionality of the input samples and to some extent reduce the training complexity. Our proposed MOS-ELM algorithm is shown in Algorithm 1. The flowchart of the MOS-ELM algorithm is shown in Figure 1.

**Uncalibrated vision servo system based on multivariate adaptive regression spline and OS-ELM**

The IBVS system controls the end effector of the manipulator to move from the current pose to the desired pose based on the feedback of the current characteristics and the expected characteristics error. Given \( n \) image pixel feature points \( s_i (i = 1, 2, \ldots, n) \in \mathbb{R}^{2 \times 1} \), \( s_i = [u_i, v_i]^T \), \( u, v \) is the pixel coordinate. Desired image feature points \( s_{di} (i = 1, 2, \ldots, n) \in \mathbb{R}^{2 \times 1} \), then the image feature point error is defined as

\[
e = S - S^*
\]  
(10)

where \( S = [s_1 \ s_2 \ \cdots \ s_n] \) is the current image feature and \( S^* = [s_{d1} \ s_{d2} \ \cdots \ s_{dn}] \) is the desired image feature. Per the definition of the interaction matrix, the linear relationship between the change in image feature error and camera speed can be obtained as follows

\[
\dot{e} = L_c v_C
\]  
(11)

Here, \( v_C = [v_x \ v_y \ v_z \ w_x \ w_y \ w_z] \) is the speed of camera, and \( L_c \) is the image Jacobian matrix. The image
Jacobian matrix corresponding to the $i$th feature point is defined as

$$L_e^i = \begin{bmatrix}
\frac{-f}{\rho_u Z} & 0 & \frac{\rho_u \bar{u} \bar{v}}{Z} & -f^2 + \rho_u^2 \bar{u}^2 & -\frac{\rho_u \bar{v}}{f} & -\bar{v} \\
0 & \frac{-f}{\rho_v Z} & \frac{\bar{v} f^2 + \rho_v^2 \bar{v}^2}{Z} & -\rho_v \bar{u} \bar{v} & -\frac{\rho_v \bar{u} \bar{v}}{f} & -\bar{u}
\end{bmatrix}$$ \( \tag{12} \)

where $(u_0, v_0)$ is the main point, $\bar{u} = u - u_0$, $\bar{v} = v - v_0$ is the pixel coordinate relative to the main point, $\rho_u, \rho_v$ is the width and height of each pixel, $Z$ is the depth of the feature point, and $f$ is the focal length of the camera. Performing an exponential decoupling decline on error ($\dot{e} = -\lambda e$) from equation (11), we have

$$v_C = -\lambda L_e^i e \quad \tag{13}$$

However, during the operation of IBVS system, the acquisition of depth information requires good camera calibration, and the acquisition process is cumbersome. In addition, in the process of calculating Jacobian matrix, image feature points $[u, v]$ are easily disturbed by external noise, so that the extracted image feature points contain external noise. The image feature points disturbed by noise can be expressed as $[u, v] = [u + u_e, v + v_e]$, $u_e, v_e$ is the error of extracting feature points under external interference. In this way, the interference of external noise makes the acquired image feature error larger, which further leads to the large difference between the acquired image Jacobian matrix and the actual Jacobian matrix, resulting in the failure of convergence and instability of IBVS system. In this article, MOS-ELM is used to predict the product of Jacobian matrix pseudo inverse and image feature error, which will avoid the process of calculating Jacobian matrix and Jacobian matrix pseudo inverse, do not need depth information and good calibration, and make the system robust to external interference. Therefore, MOS-ELM is used to predict $L_e^i e$ will not be affected by calibration error, target depth information, and external noise, and the system will not be unstable.

The IBVS system based on MOS-ELM is shown in Figure 2. MOS-ELM is used to realize the mapping

![IBVS system based on MOS-ELM](image-url)
between feature errors $e$ and $L_{ee}$, which improves the performance of the IBVS system. Through the trained MOS-ELM model, the product of the Moore–Penrose of the image Jacobian matrix and feature error can be calculated based on the real-time feature error, thus avoiding the estimation of the Jacobian matrix and its Moore–Penrose, and relieving the dependence of the IBVS system on other external information.

The flow of the IBVS system based on MOS-ELM is described as follows. First, the current image feature error is obtained through the desired image feature and the current image feature. Then, the current $L_{ee}$ is estimated using the MOS-ELM regression model trained with Algorithm 1. Subsequently, we obtain the speed of the end effector through the speed controller given by equation (13). After the speed of end effector is obtained, the joint speed of the manipulator is obtained according to the manipulator Jacobi matrix model calculation, which controls the movement of the manipulator. Finally, the image feature of the next moment is obtained from the camera imaging model, and the difference between it and the expected feature is recalculated until the feature error converges and the loop of visual servo ends.

### Experimental results and analysis

#### Experimental conditions

To evaluate the ability of our algorithm in dealing with regression problem, the proposed algorithm is performed on machine learning data sets and compared with the MARS, ELM, OS-ELM, and KOS-ELM algorithms. Then, the MOS-ELM algorithm is used in the visual servo control, and the relevant experiments are performed with the PUMA 560 six-degree-of-freedom manipulator as a simulation model. All experiments were performed on a

| Data sets                  | Algorithm (activation function) | Training time (s) | Training RMSE (var) | Testing RMSE (var) |
|----------------------------|----------------------------------|-------------------|---------------------|-------------------|
| California Housing         | MARS                             | 8.1421            | 0.1520 (0.0031)     | 0.1541 (0.0042)   |
|                            | ELM (sigmoid)                    | 0.1884            | 0.2307 (0.0023)     | 0.2321 (0.0039)   |
|                            | ELM (RBF)                        | 0.2737            | 0.2354 (0.0023)     | 0.2408 (0.0039)   |
|                            | OS-ELM (sigmoid)                 | 2.9959            | 0.1462 (0.0014)     | 0.1484 (0.0018)   |
|                            | OS-ELM (RBF)                     | 8.6241            | 0.1443 (0.0018)     | 0.1483 (0.0019)   |
|                            | KOS-ELM (sigmoid)                | 3.2987            | 0.1377 (0.0035)     | 0.1382 (0.0037)   |
|                            | KOS-ELM (RBF)                    | 10.2656           | **0.1356** (0.0028) | 0.1383 (0.0033)   |
|                            | MOS-ELM (sigmoid)                | 11.2297           | 0.1414 **(0.0013)** | 0.1387 (0.0012)   |
|                            | MOS-ELM (RBF)                    | 18.2516           | 0.1394 (0.0014)     | **0.1378** (0.0007) |
| Fried                      | MARS                             | 10.3194           | 0.1721 (0.0033)     | 0.1748 (0.0046)   |
|                            | ELM (sigmoid)                    | 0.1648            | 0.1591 (0.0014)     | 0.1603 (0.0014)   |
|                            | ELM (RBF)                        | 0.2347            | **0.1584** (0.0026) | 0.1611 (0.0032)   |
|                            | OS-ELM (sigmoid)                 | 2.0031            | 0.1748 (0.0031)     | 0.1792 (0.0029)   |
|                            | OS-ELM (RBF)                     | 5.1063            | 0.1725 (0.0052)     | 0.1721 (0.0037)   |
|                            | KOS-ELM (sigmoid)                | 9.7156            | **0.1776** (0.0009) | 0.1773 (0.0009)   |
|                            | KOS-ELM (RBF)                    | 22.8297           | 0.1689 (0.0056)     | 0.1670 (0.0056)   |
|                            | MOS-ELM (sigmoid)                | 11.2969           | 0.1592 (0.0037)     | **0.1525** (0.0055) |
|                            | MOS-ELM (RBF)                    | 42.6615           | 0.1659 (0.0060)     | 0.1543 (0.0025)   |
| Auto MPG                   | MARS                             | 2.3684            | 0.1274 (0.0102)     | 0.1301 (0.0101)   |
|                            | ELM (sigmoid)                    | 0.0037            | 0.1376 (0.0038)     | 0.1504 (0.0097)   |
|                            | ELM (RBF)                        | 0.0041            | 0.1459 (0.0074)     | 0.1662 (0.0098)   |
|                            | OS-ELM (sigmoid)                 | 0.0606            | 0.0970 (0.0049)     | 0.0983 (0.0109)   |
|                            | OS-ELM (RBF)                     | 0.1472            | 0.1075 (0.0104)     | 0.1014 (0.0273)   |
|                            | KOS-ELM (sigmoid)                | 0.1600            | 0.0932 (0.0064)     | 0.0933 (0.0146)   |
|                            | KOS-ELM (RBF)                    | 0.2109            | 0.0967 (0.0128)     | 0.0971 (0.0133)   |
|                            | MOS-ELM (sigmoid)                | 0.0266            | 0.0837 **(0.0033)** | 0.0892 (0.0093)   |
|                            | MOS-ELM (RBF)                    | 0.0906            | **0.0832** (0.0067) | **0.0854** (0.0168) |
| Metro Interstate Traffic Volume | MARS                             | 9.0106            | 0.1723 (0.0513)     | 0.1701 (0.0502)   |
|                            | ELM (sigmoid)                    | 0.2013            | 0.2197 (0.0709)     | 0.2208 (0.0791)   |
|                            | ELM (RBF)                        | 0.2305            | 0.2301 (0.0801)     | 0.2107 (0.0809)   |
|                            | OS-ELM (sigmoid)                 | 4.3019            | 0.1052 (0.0221)     | 0.1044 (0.0230)   |
|                            | OS-ELM (RBF)                     | 9.7903            | 0.0997 (0.0112)     | 0.1013 (0.0216)   |
|                            | KOS-ELM (sigmoid)                | 5.9041            | 0.0580 (0.0059)     | 0.0582 (0.0071)   |
|                            | KOS-ELM (RBF)                    | 12.1091           | 0.0501 (0.0037)     | 0.0523 (0.0040)   |
|                            | MOS-ELM (sigmoid)                | 23.9807           | 0.0502 (0.0024)     | 0.0506 (0.0030)   |
|                            | MOS-ELM (RBF)                    | 29.0105           | **0.0493** (0.0021) | **0.0500** (0.0027) |

OS-ELM: online sequential extreme learning machine; ELM: extreme learning machine; RBF: radial basis function. Boldface is the best result.
computer with a CPU frequency of 2.5 GHz and RAM of 8 GB through MATLAB2015a software. The parameter settings of the simulation experiment are given in the next section.

**Experimental parameter setting**

Our experiment is primarily based on machine learning data sets and IBVS system simulation. The algorithm parameter settings for the regression experiments using machine learning data sets are shown in Table 1. For the ELM algorithm, the hidden layer node number is the set as that shown in Table 1. The hidden layer node number and the number of initialization samples are obtained by grid optimization, where the hidden layer node number is in the range of $[5, 10, \ldots, 200]$, and the number of initialization samples is in the range of $[25, 50, \ldots, 200]$. The information for all the data sets is shown in Table 2, and the training set and test set are obtained in a random manner.

In the IBVS simulation experiments, the data set used for training was obtained through simulating the Classical IBVS system in 17 different end effector postures, and the

Figure 3. MOS-ELM predicted end effector speed compared with the actual speed. (a), (b), (c), (d), (e), and (f) are the speeds in the directions $v_x$, $v_y$, $v_z$, $w_x$, $w_y$, and $w_z$, respectively.
Figure 4. Manipulator visual servo simulation experimental results for (a) Classical IBVS, (b) ELM IBVS, (c) OS-ELM IBVS, (d) KOS-ELM IBVS, and (e) MOS-ELM IBVS, where the columns 1–3 represent feature trajectory (the red circles represent the initial features, and the blue circles represent the desired features), end effector trajectory (the red circle represents the initial pose, and the blue circle represents the end pose), and image feature error, respectively. IBVS: image-based visual servo; OS-ELM: online sequential extreme learning machine; ELM: extreme learning machine.
depth information was fixed at 2 m and the gain $\lambda = 0.5$. The coordinate of the feature point is $P$, the desired image feature is $S'$, the initial pose is $q_0$, and the error convergence threshold is 0.5. All numerical results indicate the average of the results of 50 replications of the experiment, which ensure the reliability of the experimental results.

$$P = \begin{bmatrix} 0.25 & 0.25 & -0.25 & -0.25 \\ -0.25 & 0.25 & 0.25 & -0.25 \\ 2.5 & 2.5 & 2.5 & 2.5 \end{bmatrix}$$

$$S' = \begin{bmatrix} 612 & 412 & 412 & 612 \\ 412 & 412 & 612 & 612 \end{bmatrix}$$

$$q_0 = [0 \quad \pi/4 \quad \pi/4 \quad -\pi/4] \text{rad} \quad (14)$$

### Experimental result discussion

**Regression problem experiment.** To illustrate the excellent performance of MOS-ELM in the case of regression problems, the MARS, ELM, OS-ELM, and KOS-ELM algorithms were compared with our MOS-ELM algorithm. Test the regression ability of the algorithm using machine learning data sets California Housing, Fried, Auto MPG, and Metro Interstate Traffic Volume. California Housing contains 40,768 data points for predicting housing prices for linear regression. Fried contains 10 categories of features, and its data volume is 40,768. Auto MPG is a statistical measure of the fuel consumption of the urban cycle. Metro Interstate Traffic Volume is a forecast estimate of traffic volume. All these examples show that our MOS-ELM algorithm has a high testing accuracy.

Based on the variance values of the test errors in Table 3, we found that for four data sets, the testing error variance of the algorithms MOS-ELM (RBF), MOS-ELM (RBF), MOS-ELM (sigmoid), and MOS-ELM (RBF) are 0.0007, 0.0025, 0.0093, and 0.0027, respectively. From the above data, the prediction error of our MOS-ELM algorithm for the four machine learning data sets is the smallest among all the compared algorithms. All these examples show that our MOS-ELM algorithm has good stability.

**Proposed IBVS system compared with other systems.** The predicted and actual speed comparison of the visual servo system based on the MOS-ELM algorithm for the end effector speed are shown in Figure 3. From Figure 3, four of the six components of velocity are close to reality, while the other two components fluctuate slightly. Furthermore, the sixth component converges as well as the actual velocity. The experimental results show that our MOS-ELM algorithm can effectively predict the speed of the end effector and can be applied to the vision servo system with unknown and uncalibrated depth information.

**Table 4. Performance index results of MOS-ELM compared with other algorithms.**

| Algorithm         | Convergence speed (n) | Trajectory length (m) | Accumulated error | Prediction error |
|-------------------|-----------------------|-----------------------|-------------------|------------------|
| Classical IBVS    | 261                   | 0.3906                | 9.41e+3           | —                |
| ELM IBVS          | 250                   | 0.3896                | 9.33e+3           | 8.38e-06         |
| OS-ELM IBVS       | 242                   | 0.3891                | 9.42e+3           | 6.62e-06         |
| KOS-ELM IBVS      | 233                   | 0.3867                | 9.24e+3           | 6.32e-06         |
| MOS-ELM IBVS      | 218                   | 0.3838                | 8.74e+3           | 6.16e06          |

IBVS: image-based visual servo; OS-ELM: online sequential extreme learning machine; ELM: extreme learning machine.
incremental change of the joint angle is smooth, and there is no oscillation, which indicates that this method can control the end effector movement smoothly from the initial position pose to the desired position pose.

The MOS-ELM IBVS system was compared with the Classical IBVS, IBVS based on ELM, IBVS based on OS-ELM, and IBVS based on KOS-ELM. The comparative experimental results are shown in Figure 4, and the three performance indexes are shown in Table 4. The real-time performance of different IBVS system is assessed based on the convergence speed. In addition, the small cumulative error indicates that the convergence of feature errors is faster. The shorter end effector trajectory length provides a good indication that the redundancy motion of the end effector is lesser. Furthermore, the smaller accumulated error indicates that the convergence of feature error is faster. From Figure 4, the trajectory of end effector and error convergence for several other IBVS are similar. However, it can be found from Table 4 that the iterations of the IBVS convergence based on MOS-ELM are 218, trajectory length of end effector is 0.3838 m, and error accumulation is 8.74e+3. It should be noted that these results are the smallest of all the other methods. The smallest number of iterations indicates that the IBVS algorithm based on MOS-ELM has the fastest convergence speed, which reduces the time for the manipulator to reach the expected pose. The minimum trajectory length of the end effector indicates that the IBVS algorithm based on the MOS-ELM can reduce the distance that the manipulator moves in space, while increasing the speed with which the manipulator moves to the expected position. The lower error accumulation indicates that the IBVS system based on MOS-ELM can control the error oscillation better. It can be found from Table 4, the prediction error of ELM, OS-ELM, KOS-ELM, and MOS-ELM algorithm for the manipulator data sets is 9.38e−06, 6.62e−06, 6.32e−06, and 6.16e−06, respectively. Evidently, the MOS-ELM algorithm has the smallest prediction error.

Comparison on different IBVS system configurations. To show that the method proposed in this article also has good performance in different pose configurations, here we compare the experimental results of Classical IBVS and MOS-ELM IBVS in another pose. The configuration of IBVS is as follows...
Figures 5 and 6 are the simulation results of the Classical IBVS and the MOS-ELM IBVS, respectively. Comparing the experimental results of Figures 5 and 6, MOS-ELM IBVS has a better curve in feature error convergence and manipulator joint angular velocity, and the trajectory of manipulator end effector is smoother than that of Classical IBVS. The three performance indicators of Classic IBVS and MOS-ELM IBVS are given in Table 5. The iteration times of the two methods are 277 and 137, respectively, and the trajectory length of the end effector of the Classical IBVS manipulator is much longer than that of the MOS-ELM IBVS, which further indicates that the method proposed in this article is also applicable to IBVS with different configurations.

At the same time, to further demonstrate the robustness of the MOS-ELM IBVS method proposed in this article to the noise interference of feature points, the servo simulation experiments of the Classical IBVS and MOS-ELM IBVS under noise interference are also given in this section. During the experiment, uniformly distributed random noise interference is added to all image feature points, and the interference deviation is $[-2, 2]$ pixels. Figures 7 and 8 are the experimental results of Classical IBVS and MOS-ELM IBVS under noise interference, respectively. It can be seen from Figures 7 and 8 that the Classical IBVS has

\[
P = \begin{bmatrix}
0.25 & 0.25 & -0.25 & -0.25 \\
-0.25 & 0.25 & 0.25 & -0.25 \\
1.5 & 1.5 & 1.5 & 1.5
\end{bmatrix}
\]

\[
S^* = \begin{bmatrix}
700 & 300 & 300 & 700 \\
300 & 300 & 700 & 700
\end{bmatrix}
\]

\[q_0 = [2.6524, -0.81041, 1.7506, -0.9017, 1.0816, -2.3562]\]

(15)

Table 5. Comparison of performance index results between MOS-ELM IBVS and Classical IBVS.

|                  | Classical IBVS | MOS-ELM IBVS |
|------------------|----------------|--------------|
| Convergence speed (n) | 277            | 137          |
| Trajectory length (m) | 0.3377         | 0.247        |
| Accumulated error   | 1.816e+4       | 1.01e+4      |

IBVS: image-based visual servo.
Figure 7. Experimental results of visual servo simulation of manipulator under noise interference of Classical IBVS, (a) feature trajectory (the red circles represent the initial features, and the blue circles represent the desired features), (b) end effector trajectory (the red circle represents the initial pose, and the blue circle represents the end pose), (c) image feature error, and (d) joint angular velocity. IBVS: image-based visual servo.

Figure 8. Experimental results of visual servo simulation of manipulator under noise interference of MOS-ELM IBVS, (a) feature trajectory (the red circles represent the initial features, and the blue circles represent the desired features), (b) end effector trajectory (the red circle represents the initial pose, and the blue circle represents the end pose), (c) image feature error, and (d) joint angular velocity. IBVS: image-based visual servo.
shown poor performance after being disturbed by noise, and the trajectory of its end effector has become very unsmooth. Moreover, the motion of the joint angle of the manipulator has a large amplitude jitter, which shows that the Classical IBVS has poor robustness to noise interference. In contrast, the end effector of the MOS-ELM IBVS is relatively smooth, and the joint angle of the manipulator does not have a large jitter. The performance indicators of the two methods are given in Table 6. From Table 6, the lengths of the end effector trajectories of the Classical IBVS and MOS-ELM IBVS are 0.49024 m and 0.25547 m, respectively. The lengthening of the classic IBVS end effector trajectory is mainly caused by the jitter of the robotic arm. In addition, the total error overhead of MOS-ELM IBVS is smaller than that of Classical IBVS, which further indicates that using MOS-ELM to predict the product of pseudo inverse of image Jacobian matrix and image feature error can improve the robustness to noise interference and ensure the performance of IBVS control.

Conclusion

In this study, a regression algorithm (MOS-ELM) based on MARS and OS-ELM is proposed. First, the superiority of the MOS-ELM in regression problems is verified using machine learning data sets, which is then compared with MARS, ELM, OS-ELM, and KOS-ELM. Then, an uncalibrated IBVS system based on MOS-ELM is proposed and compared with the Classical IBVS, ELM IBVS, OS-ELM IBVS, and KOS-ELM IBVS. It is observed that the proposed algorithm solves the problem of accurate image Jacobian prediction and improves the robustness of the system to noise disturbances. The proposed method uses the MARS method to assess the importance of the features in the training samples and selects more important samples as the input features for the OS-ELM. By combining these two aspects, the feature dimensionality can be reduced, consequently reducing the computational complexity. The improved performance of the IBVS system with our MOS-ELM algorithm is proved by the performance analysis of three performance indexes including convergence speed, trajectory length, and cumulative error. Experimental results show that our algorithm is feasible and performs well when applied to IBVS systems.

Table 6. Comparison of performance index results of MOS-ELM IBVS and Classical IBVS under noise interference.

|                      | Classical IBVS | MOS-ELM IBVS |
|----------------------|----------------|--------------|
| Convergence speed (n)| 277            | 137          |
| Trajectory length (m)| 0.49024        | 0.25547      |
| Accumulated error     | 1.8656e±4      | 1.4452e±4    |

IBVS: image-based visual servo.

Declaration of conflicting interests

The author(s) declared no potential conflicts of interest with respect to the research, authorship, and/or publication of this article.

Funding

The author(s) disclosed receipt of the following financial support for the research, authorship, and/or publication of this article: This work was supported by Key R&D Program of Zhejiang Province (No. 2021C03013) and Zhejiang Provincial Natural Science Foundation of China (No. LZ20F020003).

ORCID iD

Zhiyu Zhou https://orcid.org/0000-0003-4487-8192

Supplemental video

Supplemental video for this article is available online.

References

1. Zhou Z, Guo H, Wang Y, et al. Inverse kinematics solution for robotic manipulator based on extreme learning machine and sequential mutation genetic algorithm. Int J Adv Robot Syst 2018; 15(4): 1729881418792992.
2. Zhou Z and Wu B. Adaptive sliding mode control of manipulators based on fuzzy random vector function links for friction compensation. Optik 2021; 227: 166055.
3. Wang R, Zhang X, and Fang Y. Visual tracking of mobile robots with both velocity and acceleration saturation constraints. Mech Syst Signal Process 2021; 150: 107274.
4. Shi H, Xu M, and Hwang KS. A fuzzy adaptive approach to decoupled visual servoing for a wheeled mobile robot. IEEE Trans Fuzzy Syst 2020; 28(12): 3229–3243.
5. Chang W C. Robotic assembly of smartphone back shells with eye-in-hand visual servoing. Robot Comput Integr Manuf 2018; 50: 102–113.
6. Sha X, Li H, Li W, et al. Three-dimensional positioning control based on stereo microscopic visual servoing system. Opt Eng 2015; 54(1): 013106.
7. Gu J, Wang H, Pan Y, et al. Neural network based visual servo control for CNC load/unload manipulator. Optik 2015; 126(23): 4489–4492.
8. Song X and Miaomiao F. CLFs-based optimization control for a class of constrained visual servoing systems. Int J Adv Robot Syst 2017; 67: 507–514.
9. Wang Y, Lang H, and De Silva CW. A hybrid visual servo controller for robust grasping by wheeled mobile robots. IEEE/ASME Trans Mechatron 2010; 15(5): 757–769.
10. Zhou L, Li H, Zhao W, et al. Imaged-based visual servo control for a VTOL aircraft. Math Prob Eng 2017; 2017: 4806769.
11. Zhao Y M, Lin Y, Xi F, et al. Switch-based sliding mode control for position-based visual servoing of robotic riveting system. J Manuf Sci Eng 2017; 139(4): 041010.
12. Li B, Fang Y, and Zhang X. 2D trifocal tensor based visual servo regulation of nonholonomic mobile robots. Zidonghua Xuebao/Acta Autom Sin 2014; 40(12): 2706–2715.
13. Mahony R and Hamel T. Image-based visual servo control of aerial robotic systems using linear image features. *IEEE Trans Robot* 2005; 21(2): 227–239.
14. Nadeau C, Krupa A, Petr J, et al. Moments-based ultrasound visual servoing: from a mono-to multiplane approach. *IEEE Trans Robot* 2016; 32(6): 1558–1564.
15. Copot C, Lazar C, and Burlacu A. Predictive control of non-linear visual servoing systems using image moments. *IET Control Theory Appl* 2012; 6(10): 1486–1496.
16. Wang J and Cho H. Micropeg and hole alignment using image moments based visual servoing method. *IEEE Trans Ind Electron* 2008; 55(3): 1286–1294.
17. Liu YH, Wang H, Chen W, et al. Adaptive visual servoing using common image features with unknown geometric parameters. *Automatica* 2013; 49(8): 2453–2460.
18. Kang M, Chen H, and Dong J. Adaptive visual servoing with an uncalibrated camera using extreme learning machine and Q-learning. *Neurocomputing* 2020; 402: 384–394.
19. Qiu Z and Wu Z. Adaptive neural network control for image-based visual servoing of robot manipulators. *IET Control Theory Appl* 2022; 16(4): 443–453.
20. Gao Z, Ma C, Zhang J, et al. Remaining useful life prediction of integrated modular avionics using ensemble enhanced online sequential parallel extreme learning machine. *Int J Mach Learn Cybern* 2021; 12(7): 1893–1911.
21. Peng X, Wang B, Zhang L, et al. Adaptive online sequential extreme learning machine with kernels for online ship power prediction. *Energies* 2021; 14(17): 5371.
22. Polat Ö and Kayhan SK. GPU-accelerated and mixed norm regularized online extreme learning machine. *Concurr Comput Pract Exp* 2022; 34(15): e6967. doi: 10.1002/cpe.6967.
23. Kao LJ and Chiu CC. Application of integrated recurrent neural network with multivariate adaptive regression splines on SPC-EPC process. *J Manuf Syst* 2020; 57: 109–118.
24. Li K, Sun Y, Robinson D, et al. A new strategy to benchmark and evaluate building electricity usage using multiple data mining technologies. *Sustain Energy Technol Assess* 2020; 40: 100770.
25. Depren SK and Kartal M T. Prediction on the volume of non-performing loans in Turkey using multivariate adaptive regression splines approach. *Int J Finance Econ* 2021; 26(4): 6395–6405.
26. Bose A, Hsu CH, Roy SS, et al. Forecasting stock price by hybrid model of cascading multivariate adaptive regression splines and deep neural network. *Comput Electr Eng* 2021; 95: 107405.
27. Cartocci N, Napolitano MR, Crocetti F, et al. Data-driven fault diagnosis techniques: non-linear directional residual vs. machine-learning-based methods. *Sensors* 2022; 22(7): 2635.
28. Hoang ND, Chen CT, and Liao KW. Prediction of chloride diffusion in cement mortar using multi-gene genetic programming and multivariate adaptive regression splines. *Measurement* 2017; 112: 141–149.
29. Borodin V, Bourtembourg J, Hnaien F, et al. Predictive modeling with panel data and multivariate adaptive regression splines: case of farmers crop delivery for a harvest season ahead. *Stoch Environ Res Risk Assess* 2016; 30(1): 309–325.
30. Nian R, He B, and Lendasse A. 3D object recognition based on a geometrical topology model and extreme learning machine. *Neural Comput Appl* 2013; 22(3): 427–433.
31. Kumar A, Paramesran R, Lim CL, et al. Tchebichef moment based restoration of Gaussian blurred images. *Appl Opt* 2016; 55(32): 9006–9016.
32. Zhang YD, Zhao G, Sun J, et al. Smart pathological brain detection by synthetic minority oversampling technique, extreme learning machine, and Jaya algorithm. *Multimedia Tools Appl* 2018; 77(17): 22629–22648.
33. Nayak PK, Mishra S, Dash PK, et al. Comparison of modified teaching–learning-based optimization and extreme learning machine for classification of multiple power signal disturbances. *Neural Comput Appl* 2016; 27(7): 2107–2122.
34. Ertugrul ÖF and Altun Ş. Developing correlations by extreme learning machine for calculating higher heating values of waste frying oils from their physical properties. *Neural Comput Appl* 2017; 28(11): 3145–3152.
35. Liang NY, Huang GB, Saratchandran P, et al. A fast and accurate online sequential learning algorithm for feedforward networks. *IEEE Trans Neural Network* 2006; 17(6): 1411–1423.
36. Yadav B, Ch S, Mathur S, et al. Discharge forecasting using an online sequential extreme learning machine (OS-ELM) model: a case study in Neckar River, Germany. *Measurement* 2016; 92: 433–445.
37. Huang GB, Zhou H, Ding X, et al. Extreme learning machine for regression and multiclass classification. *IEEE Trans Syst Man Cybern Pt B (Cybern)* 2011; 42(2): 513–529.
38. Nobrega JP and Oliveira ALI. Kalman filter-based method for online sequential extreme learning machine for regression problems. *Eng Appl Artif Intell* 2015; 44: 101–110.
39. Dua D and Graff C. (2019). *UCI Machine learning repository*. Irvine, CA: University of California, School of Information and Computer Science. (accessed August 8, 2021) http://archive.ics.uci.edu/ml.