2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)
October 14-18, 2024. Abu Dhabi, UAE

A Cascaded Broad Learning System for Manipulator Motion Control

Guoyu Zuo[1,2,*], Shuaifeng Dong[1,2], Jiyong Zhou[1,2] and Shuangyue Yu[1,2]

Abstract— Intelligent control methods have led to a significant simplification of the robotic arm modeling and control tuning process, and thus they have been widely used. To further improve the precision of robotic arm motion control, this paper proposes a robotic arm motion control strategy based on a cascaded feature-enhanced elastic-net broad learning system (CFE-EN-BLS). This will fully extract data features to improve motion control accuracy. Moreover, ElasticNet regression is introduced to reduce feature redundancy. Finally, Lyapunov stability theory is introduced to constrain the learning parameters of the proposed learning method to enhance the convergence of the control strategy. The simulation and experiment show that the proposed control strategy can realize high-precision trajectory tracking control of the robotic arm.

I. INTRODUCTION

Robotic arm has been widely used in industrial production, agricultural production, aerospace, medical and surgery, service and education, and various other domains [1]–[5]. In recent years, there has been a significant shift in the requirements for robotic arms and their control systems, with greater emphasis now being placed on reliability, precision, and safety in diverse workspaces and tasks [6]–[8]. Therefore, it has become an important research topic in the field of robotic arm motion control to design and implement a high-performance motion control strategy.

Learning-based robotic arm motion control strategies can overcome the limitations of traditional control strategies relying on accurate mathematical modeling lacking generalization capabilities [9]–[12]. Traditional learning-based robotic arm motion control strategies are mainly based on machine learning and shallow neural networks [13]–[16]. However, those strategies are easy to fall into the limitations of local minima and sensitive learning rate selection, which affect the accuracy of the robotic arm motion control strategies. To solve the above problems, some scholars have introduced deep learning and reinforcement learning into the field of robotic arm motion control and designed numerous high-performance motion control strategies [17]–[19]. However, these control strategies require a complicated training process to determine the control learning parameters, consuming a huge amount of computing resources.

The broad learning system (BLS) control strategy can significantly simplify the model training process and reduce the consumption of computational resources by virtue of its simple and flexible network structure and accurate approximation performance, as well as its good generalization [20]. Some scholars have attempted to apply it to the field of robotic arm motion control [21], [22]. Huang et al. combined the BLS with deterministic learning and proposed a new type of neuro-adaptive control for the realization of robotic arm motion generalization learning [23]. They further combined the BLS with a fuzzy neural network and proposed a broad fuzzy neural network control strategy, which realized robotic arm ideal tracking and interaction performance under the condition of ensuring the stability of the closed-loop system [24]. Xu et al. applied the BLS to micro-robot control for the first time and combined the learning algorithm with Lyapunov theory to get the controller parameter constraints obtaining robust generalization capability and stable convergence of errors [25]. Zuo et al. proposed a robotic control strategy based on the incremental Bayesian fuzzy broad learning system (IBFBLS), which improved robot joints' servo control accuracy [26]. However, due to the sparsity of the BLS, there is the problem of inadequate feature extraction [27] thus leading to the problem of insufficient control accuracy. Furthermore, these control strategies ignore the impact of joint control accuracy on the overall system performance.

To address the above problems, this paper comprehensively considers the advantages of the cascade mechanism and ElasticNet learning to design a robot arm motion control strategy with sufficient feature extraction, low computational redundancy, and high control accuracy. Compared to existing methods, the main contributions are as follows:

- The CFE-EN-BLS-based control strategy is proposed and applied to the motion control of a robotic arm, which significantly improves the control accuracy.
- A cascaded structure is designed to improve the ability of the broad learning network to extract valuable information. Meanwhile, the proposed strategy is adapted by combining the elastic-net regression, which is feature selection for cascaded feature-enhanced broad learning system (CFE-BLS) mapping features thereby reducing computational

[1] Faculty of Information Technology, Beijing University of Technology, Beijing 100124, China.
[2] Beijing Key Laboratory of Computing Intelligence and Intelligent Systems, Beijing 100124, China.
* Corresponding author. E-mail address: zuoguoyu@bjut.edu.cn
This work was supported by the National Natural Science Foundation of China (62373016), the Open Projects Program of State Key Laboratory of Multimodal Artificial Intelligence Systems (MAIS-2023-22).

redundancy and improving the efficiency of the control method.
- The proposed motion control strategy is combined with Lyapunov's theory to obtain constraints on the learning parameters, which provide a stable convergence capability for the control strategy

II. PROBLEM FORMULATION AND PRELIMINARIES

A. Robotic Arm Motion Control

Define the robotic arm joint angle, angular velocity, and angular acceleration as $\boldsymbol{\theta} = [\theta_1, \theta_2, \ldots, \theta_N], \dot{\boldsymbol{\theta}} = [\dot{\theta_1}, \dot{\theta_2}, \ldots, \dot{\theta_N}], \ddot{\boldsymbol{\theta}} = [\ddot{\theta_1}, \ddot{\theta_2}, \ldots, \ddot{\theta_N}]$, then the robotic arm dynamics model can be expressed as follows:

$$M(\boldsymbol{\theta})\ddot{\boldsymbol{\theta}} + C(\boldsymbol{\theta})\dot{\boldsymbol{\theta}} + G(\boldsymbol{\theta}) + \boldsymbol{\tau}_n = \boldsymbol{\tau}_c \tag{1}$$

where $M(\boldsymbol{\theta})$ denotes the inertia matrix of the robotic arm dynamics model. $C(\boldsymbol{\theta})$ denotes the centrifugal and Koch force vectors, $G(\boldsymbol{\theta})$ denotes the gravity term, $\boldsymbol{\tau}_n$ denotes the perturbation suffered by the robotic arm during its motion, and $\boldsymbol{\tau}_c$ denotes the robotic arm joint control moment.

Define deviations from ideal trajectories in Cartesian space as shown in the following equation:

$$\boldsymbol{e}(t) = \boldsymbol{\theta}(t) - \boldsymbol{\theta}_r(t) \tag{2}$$

where $\boldsymbol{e}(t)$ denotes the angular error, $\boldsymbol{\theta}(t), \boldsymbol{\theta}_r(t)$ denotes the actual angle, and the desired angle, respectively.

To ensure the control accuracy of the robotic arm motion control, this section considers constructing the robotic arm motion control based on the first-order motion control system. Therefore the motion error based on velocity level can be expressed as:

$$\dot{\boldsymbol{e}}(t) = \dot{\boldsymbol{\theta}}(t) - \dot{\boldsymbol{\theta}}_r(t) \tag{3}$$

where $\dot{\boldsymbol{\theta}}_r(t)$ denotes the desired angular velocity and $\dot{\boldsymbol{\theta}}(t)$ denotes the actual angular velocity, which is obtained in this paper by calculating Eq. (4):

$$\theta(t) \approx \theta(t-1) + \dot{\theta}(t-1)\Delta t \tag{4}$$

where Δt denotes the time interval. When Δt is set to very small, " $\approx$ " is approximated as " $=$ ".

This section specifies a goal function aimed at achieving zero tracking error, as shown in Eq. (5):

$$f_\theta(\boldsymbol{e}) = \min_{\theta(t)}(\dot{\boldsymbol{\theta}}(t) - \dot{\boldsymbol{\theta}}_r(t)) \tag{5}$$

Approximating the desired velocity with the output of the motion control strategy will facilitate the convergence of the velocity-tracking deviation $\dot{\boldsymbol{e}}$ to 0. This is a key factor in ensuring that the robotic arm accurately tracks the desired trajectory. Therefore, the issue of achieving precise position tracking in robotic arm motion control is considered equivalent to solving an optimization problem.

B. Cascade Feature-enhanced Broad Learning System

A cascaded feature-enhanced broad learning system (CFEBLS) is designed in this section for robotic arm motion control.

1) Cascade Feature Node Generation: Define the training sample dataset $\{(\mathbf{X}, \mathbf{Y}) \mid \mathbf{X} \in R^{N\times d}, \mathbf{Y} \in R^{N\times c}\}$, where N denotes the number of input samples, and d denotes the dimensionality of each sample. $\mathbf{Y}$ denotes the target output corresponding to the input sample $\mathbf{X}$, and c denotes the target output dimensionality.

Next, defining the feature cascade matrix $\mathbf{A}_0 = \varnothing$, $\mathbf{A}_1 = [\mathbf{A}_0, \mathbf{X}]$ and $\mathbf{A}_i = [\mathbf{X}, \mathbf{Z}_{i-1}]$ built from the combination of the input data and the $(i-1)$th group of feature nodes, then the mapped i-th group of feature nodes can be described as:

$$\mathbf{Z}_i = \Phi_i(\alpha_{ei}\mathbf{A}_i + \beta_{ei}) \quad i = 1, 2, \ldots, n \tag{6}$$

where α_{ei} and β_{ei} denote the weights and biases of the feature nodes, respectively, both of which are randomly generated. Φ_i denotes the function used to activate the feature nodes.

The obtained node features are given in the following equation:

$$\mathbf{Z}^n = [\mathbf{Z}_1, \mathbf{Z}_2, \ldots, \mathbf{Z}_n] \tag{7}$$

2) Cascade Enhancement Node Generation: To enhance the nodes by the cascade, this section defines the new enhancement nodes that are generated from the feature nodes and the previous enhancement nodes.

Defining the enhancement cascade matrix $\mathbf{B}_1 = \mathbf{Z}^n$, $\mathbf{B}_2 = [\mathbf{Z}^n, \mathbf{H}_1]$ and $\mathbf{B}_j = [\mathbf{Z}^n, \mathbf{H}_{j-1}]$ consisting of the feature node $\mathbf{Z}^n$ and the $(j-1)$th group of enhancement nodes, then the jth group of enhancement nodes can be described as:

$$\mathbf{H}_j = \Psi_j(\alpha_{hj}\mathbf{B}_j + \beta_{hj}) \quad j = 1, 2, \ldots, m \tag{8}$$

where α_{hj} and β_{hj} denote the weight and bias of the augmented node, respectively, both randomly generated. Ψ_j denotes the activation function of the augmented node.

The obtained enhanced feature node layers are as follows:

$$\mathbf{H}^m = [\mathbf{H}_1, \mathbf{H}_2, \ldots, \mathbf{H}_m]. \tag{9}$$

Then the acquired node can be described as:

$$\begin{aligned}\mathbf{F}_n^m =& [\mathbf{Z}^n \mid \mathbf{H}^m] \\ =& [\mathbf{Z}_1, \mathbf{Z}_2, \ldots, \mathbf{Z}_n, \mathbf{H}_1, \mathbf{H}_2, \ldots, \mathbf{H}_m].\end{aligned} \tag{10}$$

The CFEBLS model is characterized by the following description:

$$\mathbf{Y}(X) = \sum_{i=1}^{n} \mathbf{w}_i \mathbf{Z}^n + \sum_{j=1}^{m} \mathbf{w}_j \mathbf{H}^m \tag{11}$$

where $\mathbf{w}_i$ and $\mathbf{w}_j$ denote the matrix weights of the output layer, learned through the training process of the network.

III. CFE-EN-BLS MOTION CONTROL STRATEGY

This section provides an in-depth explanation of the motion control approach utilizing the Cascaded Feature-enhanced ElasticNet Broad Learning System. The control strategy mainly consists of a motion control network based on CEF-EN-BLS and an ElasticNet regression algorithm. The overall structure of the motion strategy is shown in Fig. 1.

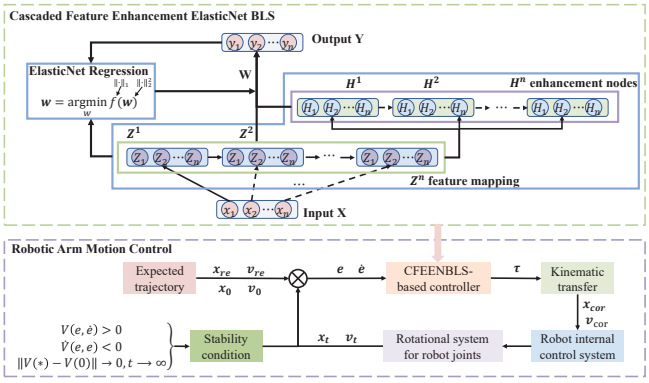


Fig. 1. Motion control strategy for robotic arm with CEF-EN-BLS.

A. Robotic Arm Motion Control Based on CFE-BLS

This section is devoted to correcting the velocity to realize the robotic arm motion control and the principle is shown as follows:

$$\dot{\theta}_{cor}(t) = \dot{\theta}_r + \dot{e}(t) \tag{12}$$

where $\dot{\theta}_{cor}(t)$ denotes the corrected velocity.

A reasonable transfer function is constructed to represent the transfer relationship between the corrected velocity and the corrected position. The principle is expressed as follows:

$$\theta_{cor}(t) \approx \theta(t-1) + \dot{\theta}_{cor}(t-1)\Delta t \tag{13}$$

where $\dot{\theta}_{cor}(t-1)$ and $\theta_{cor}(t)$ denote the corrected velocity and position, respectively.

Define the robotic arm motion control strategy as:

$$\tau = \dot{\boldsymbol{e}} = f_\theta(\boldsymbol{e}). \tag{14}$$

The inputs and outputs based on the learner are position deviation and corrected velocity, respectively. Then the CFE-BLS-based robotic arm motion control strategy is denoted as:

$$\tau = \sum_{i=1}^{n} \mathbf{w}_i \Phi_i(\alpha_{ei}\mathbf{A}_i + \beta_{ei}) + \sum_{j=1}^{m} \mathbf{w}_j \Psi_j(\alpha_{hj}\mathbf{B}_j + \beta_{hj}) \tag{15}$$

where $\mathbf{w}_i$ denotes the output weight of the feature node. $\mathbf{w}_j$ denotes the output weight of the enhancement node.

Φ, Ψ denote the feature node activation function, and enhancement node activation function, respectively, as shown in Eq. (16).

$$\begin{cases} \Phi = \frac{1}{1+exp(-\Theta)} - \frac{1}{2} \\ \Psi = \frac{1}{1+exp(-\Xi)} - \frac{1}{2} \end{cases} \tag{16}$$

where $\Theta = \alpha_{ei}\mathbf{A}_i + \beta_{ei}$ and $\Xi = \alpha_{hj}\mathbf{B}_j + \beta_{hj}$, the activation functions are all continuously differentiable. and the following conditions are satisfied:

$$\begin{cases} \Phi(0) = 0, \quad \dot{\Phi}(\Theta) > 0, \\ \Psi(0) = 0, \quad \dot{\Psi}(\Xi) > 0. \end{cases} \tag{17}$$

The training of the model entails solving an optimization task to ensure good approximation capability of the designed controller, i.e.

$$\min_{\mathbf{w}} \sum_{k=1}^{M} \| \sum_{i=1}^{n} \mathbf{w}_i \Phi_i(\Theta) + \sum_{j=1}^{m} \mathbf{w}_j \Psi_j(\Xi) - \tau_m^{de} \| \tag{18}$$

where τ_m^{de} denotes the desired output of the control strategy.

B. ElasticNet Regression Algorithm

To reduce the complexity of the CFE-BLS model to prevent overfitting, ElasticNet regression is introduced in this subsection.

Considering the presence of unknown perturbations during the movement of the robotic arm, the output of the CFE-EN-BLS network is represented in this section as:

$$\hat{\mathbf{Y}} = f_W(\mathbf{W}^T \mathbf{F}_n^m(\mathbf{X})) + \varepsilon \tag{19}$$

where ε denotes random noise, which is defined as Gaussian white noise for ease of computation.

Define the cost function as:

$$\begin{aligned} f_{cost}(\mathbf{w}) = & \sum_{k=1}^{N} (\hat{y}_k - \mathbf{w}^T \mathbf{F}_n^m(\mathbf{X}))^2 \\ & + \lambda\rho\|w_k\|_1 + \frac{\lambda(1-\rho)}{2}\|w_k\|_2^2 \end{aligned} \tag{20}$$

where λ is the regularization coefficient, which represents the strength of the total regularization, larger values of which have a stronger regularization effect, resulting in more coefficients converging to zero, leading to sparser models, and smaller values of λ allowing the model to fit more complex relationships. ρ denotes the proportion of the L_1 paradigm in the regularization. Thus, Elastic-Net regression combines the feature selection capability of Lasso regression with the multicollinearity handling capability of Ridge regression.

Determine the desired network weights by solving for the weights that minimize the cost function, i.e., solve the following optimization problem:

$$\mathbf{w} = \arg\min_{w} f_{cost}(\mathbf{w}). \tag{21}$$

This objective function minimization optimization problem can be solved by optimization algorithms such as coordinate descent and gradient descent to find the best weight parameters.

C. Stability Analysis

Drawing upon the principle of Lyapunov stability analysis, the control strategy constructed need to guarantee the system's global asymptotic stability, with the necessary condition being the fulfillment of Eq. (22):

$$\begin{cases} V(e,\dot{e}) > 0 \\ \dot{V}(e,\dot{e}) < 0 \\ V(e^*,\dot{e}^*) = 0 \\ \|V(e,\dot{e}) - V(e^*,\dot{e}^*)\| \to 0, t \to \infty \end{cases} \tag{22}$$

The system is globally asymptotically stabilized when the Lyapunov function satisfies the above equation and the system state e, $\dot{e}$ converges to the steady state point $[e^*,\dot{e}^*] = 0$.

The Lyapunov-candidate-function (LCF), denoted as $V(e,\dot{e})$, is designed as

$$V = \frac{1}{2}e^T e + \frac{1}{2}\dot{e}^T\dot{e} > 0. \tag{23}$$

To satisfy Eq. (22), it is only necessary to make the derivative of the LCF less than zero, i.e.

$$\dot{V} = e^T\dot{e} + \dot{e}^T\ddot{e} < 0. \tag{24}$$

Next, the stability constraints of the control strategy need to be obtained by deriving the derivative $\dot{V}$of the LCF.

In Equation (23), while e, $\dot{e}$ and $\ddot{e}$ are mentioned, it is noted that $\ddot{e}$ does not directly appear in the above context. Instead, The designed motion control network model employs e as its input and $\dot{e}$ as its output. Therefore, this subsection focuses on deriving the expression for $\dot{e}$ utilizing Equation (15) as a starting point. Following this, the expression for $\ddot{e}$ is subsequently computed, although it was initially stated to be absent in the immediate context.

$$\begin{aligned} \dot{\boldsymbol{e}} = & \sum_{i=1}^{n} \mathbf{w}_{ei}\Phi_i(\alpha_{ei}\boldsymbol{e} + \beta_{ei}) \\ & + \sum_{i=1}^{n} \mathbf{w}_{ei}\Phi_i(\alpha_{ei}\mathbf{Z}_{i-1} + \beta_{ei}) \\ & + \sum_{j=1}^{m} \mathbf{w}_{hj}\Psi_j(\alpha_{hj}\mathbf{Z}_i + \beta_{hj}) \\ & + \sum_{j=1}^{m} \mathbf{w}_{hj}\Psi_j(\alpha_{hj}\mathbf{H}_{j-1} + \beta_{hj}) \end{aligned} \tag{25}$$

Based on Eq. (25), $\ddot{e}$ is computed as

$$\begin{aligned} \ddot{e} = & \sum_{i=1}^{n} \frac{d[\Phi_i(\alpha_{ei}\boldsymbol{e} + \beta_{ei})]}{d(\alpha_{ei}\boldsymbol{e} + \beta_{ei})}\mathbf{w}_{ei}\alpha_{ei}\dot{\boldsymbol{e}} \\ & + \sum_{i=1}^{n} \frac{d[\mathbf{w}_{ei}\Phi_i(\alpha_{ei}\mathbf{Z}_{i-1} + \beta_{ei})]}{d(\alpha_{ei}\mathbf{Z}_{i-1} + \beta_{ei})}\mathbf{w}_{ei}\alpha_{ei}\dot{\mathbf{Z}}_{i-1} \\ & + \sum_{j=1}^{m} \frac{d[\mathbf{w}_{hj}\Psi_j(\alpha_{hj}\mathbf{Z}_i + \beta_{hj})]}{d(\alpha_{hj}\mathbf{Z}_i + \beta_{hj})}\mathbf{w}_{hj}\alpha_{hj}\dot{\mathbf{Z}}_i \\ & + \sum_{j=1}^{m} \frac{d[\mathbf{w}_{hj}\Psi_j(\alpha_{hj}\mathbf{H}_{j-1} + \beta_{hj})]}{d(\alpha_{hj}\mathbf{H}_{j-1} + \beta_{hj})}\mathbf{w}_{hj}\alpha_{hj}\dot{\mathbf{H}}_{j-1}. \end{aligned} \tag{26}$$

In (17), the derivatives of the activation functions are all strictly positive, meaning they are all greater than zero. These positive derivative terms in Eq. (26) are denoted as

$$\begin{aligned} & \frac{d[\Phi_i(\alpha_{ei}\boldsymbol{e} + \beta_{ei})]}{d(\alpha_{ei}\boldsymbol{e} + \beta_{ei})} = \frac{d\Phi(\triangle_i)}{d\triangle_i} > 0, \\ & \frac{d[\mathbf{w}_{ei}\Phi_i(\alpha_{ei}\mathbf{Z}_{i-1} + \beta_{ei})]}{d(\alpha_{ei}\mathbf{Z}_{i-1} + \beta_{ei})} = \frac{d\Phi(\square_i)}{d\square_i} > 0, \\ & \frac{d[\mathbf{w}_{hj}\Psi_j(\alpha_{hj}\mathbf{Z}_i + \beta_{hj})]}{d(\alpha_{hj}\mathbf{Z}_i + \beta_{hj})} = \frac{d\Psi(\lozenge_j)}{d\lozenge_j} > 0, \\ & \text{and } \frac{d[\mathbf{w}_{hj}\Psi_j(\alpha_{hj}\mathbf{H}_{j-1} + \beta_{hj})]}{d(\alpha_{hj}\mathbf{H}_{j-1} + \beta_{hj})} = \frac{d\Psi(\heartsuit_j)}{d\heartsuit_j} > 0. \end{aligned}$$

Next, by substituting the expressions from (25) and (26) into (24), we obtain a further derivation of Eq. (24).

$$\begin{aligned} \dot{V} = & \boldsymbol{e}^T \Bigg[\sum_{i=1}^{n} \mathbf{w}_{ei}\Phi_i(\alpha_{ei}\boldsymbol{e} + \beta_{ei}) \\ & + \sum_{i=1}^{n} \mathbf{w}_{ei}\Phi_i(\alpha_{ei}\mathbf{Z}_{i-1} + \beta_{ei}) \\ & + \sum_{j=1}^{m} \mathbf{w}_{hj}\Psi_j(\alpha_{hj}\mathbf{Z}_i + \beta_{hj}) \\ & + \sum_{j=1}^{m} \mathbf{w}_{hj}\Psi_j(\alpha_{hj}\mathbf{H}_{j-1} + \beta_{hj}) \Bigg] \\ & + \dot{e}^T \Bigg[\sum_{i=1}^{n} \frac{d\Phi(\triangle_i)}{d\triangle_i}\mathbf{w}_{ei}\alpha_{ei}\dot{\boldsymbol{e}} \\ & + \sum_{i=1}^{n} \frac{d\Phi(\square_i)}{d\square_i}\mathbf{w}_{ei}\alpha_{ei}\dot{\mathbf{Z}}_{i-1} \\ & + \sum_{j=1}^{m} \frac{d\Psi(\lozenge_j)}{d\lozenge_j}\mathbf{w}_{hj}\alpha_{hj}\dot{\mathbf{Z}}_i \\ & + \sum_{j=1}^{m} \frac{d\Psi(\heartsuit_j)}{d\heartsuit_j}\mathbf{w}_{hj}\alpha_{hj}\dot{\mathbf{H}}_{j-1} \Bigg]. \end{aligned} \tag{27}$$

Since the proposed motion control strategy designs a broad learning network in a cascade manner, after derivation and analysis, it should also satisfy $\mathbf{w}_{ei}\prod_{i=1}^{n}\alpha_{ei} < 0$, $\mathbf{w}_{hj}\prod_{j=1}^{m}\alpha_{hj} < 0$. In this way, ①+② will be negative. As mentioned earlier, the derivative and slope of the activation function are positive. For ensuring that the system achieves global asymptotic stability, which implies that the system state can converge from any initial value, it is crucial that Equation (24) holds at all times. The resulting constraints derived from this analysis are

$$\beta_{ei} = 0, \quad \beta_{hj} = 0 \tag{28a}$$

$$\mathbf{w}_{ei} < 0, \quad \mathbf{w}_{hj} < 0 \tag{28b}$$

$$\alpha_{ei} > 0, \quad \alpha_{hj} > 0 \tag{28c}$$

At this point, the constraints on the stability of the system are obtained.

IV. EXPERIMENTS

A. Simulation Experiments

The simulation was run on a server with Windows 10. The runtime environment is PyCharm 2023.1.2, Python 3.7. Hardware configuration: CPU is Intel(R) Core(TM) i9-10900X@3.70GHz, GPU is RTX 3090 Ti, and RAM is 64G. The experiment employs a sliding mode control approach to follow a reference trajectory, initiating from various starting points, to acquire the necessary training demonstration data for the implementation of the control strategy. Define $y = \pi sin(t), t \in [0, 3\pi]$ as the reference trajectory of the joint. After several experiments in MATLAB, 15 tracking trajectories with good convergence are selected. These tracking trajectories run for 3πs, and the reference trajectories are shown in Fig. 2 with the 15 converged demonstration trajectories. The inputs to the control strategy are the position error and velocity error of the demonstration trajectories, and the output is the corrected velocity. The errors in position and velocity are calculated by comparing the tracking trajectories with the reference trajectories and determining the degree of convergence between them.

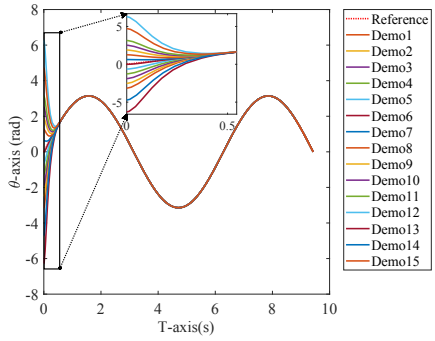


Fig. 2. Reference trajectory and demonstration trajectories.

Then, the collected demonstration data is partitioned into two sets: training and testing. The training set comprises 10 converged demonstration trajectories, while the testing set consists of 5 converged demonstration trajectories. The data are loaded into the simulation environment to train the control strategy. The comparison algorithms and ablation algorithms are broad learning system (BLS), incremental enhanced-node broad learning system (I-BLS), cascaded feature node broad learning system (CF-BLS), cascaded enhanced node broad learning system (CE-BLS), cascaded feature enhanced broad learning system (CFE-BLS), and the proposed cascaded feature-enhanced elastic-net broad learning system (CFE-EN-BLS). All control strategies have an initial feature node of 10, an initial enhancement node of 10, and a step size of 10. Considering the consistency of feature extraction, the I-BLS training rounds are set to 5, and 2 enhancement nodes are added each time. Through experimental debugging, the ElasticNet regression parameters are chosen as $\lambda = 1e^{-6}$, $\rho = 0.5$.

As an example, demonstration trajectory 8 is used for the simulation test of the joint motion control of the robotic arm, and the simulation visualization results are shown in Fig. 3. The corrected velocity is the output of the control strategy, as far as the corrected position is calculated by Eq. (13). Position deviation and velocity deviation are the differences between the control strategy real-time data and the demonstration data at the same moment.

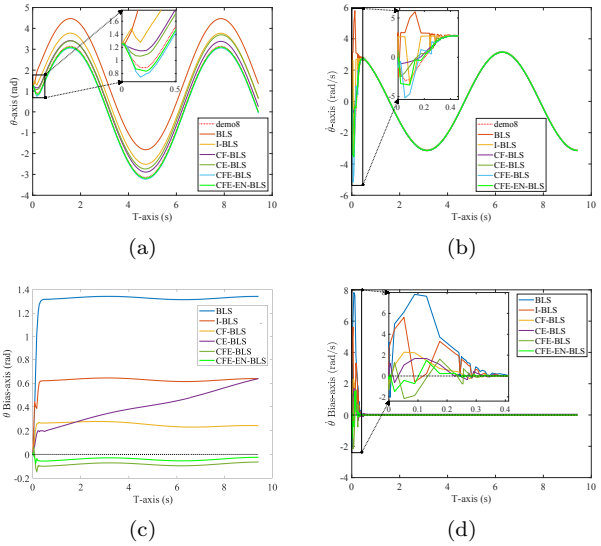


Fig. 3. Tracking results of different control strategies. (a) and (b) depict the tracking results of angular position and angular velocity with different control strategies, respectively. (c) and (d) depict the errors in angular position and angular velocity with respect to the demonstrated trajectories for different control strategies, respectively.

It is observed that the BLS exhibits tracking performance, yet it demonstrates notably large tracking errors with demonstration trajectory 8, particularly evident in Fig. 3(c). The angular position consistently diverges from the demonstration trajectory and converges to a stable state, a behavior attributed to the adjustment following abrupt changes in angular velocity. This underscores the significance of effectively extracting data features to mitigate potential errors. The incorporation of enhancement nodes into the I-BLS enhances the model's learning capacity to some extent, resulting in a reduction in angular position error. This suggests that the addition of enhancement nodes facilitates the extraction of valuable information from feature mapping. However, this augmentation also introduces redundancy in the extracted information, thereby constraining performance improvement. The corner position tracking performance of CF-BLS and CE-BLS control strategies is better than that of the BLS control strategy, which indicates that cascading

feature nodes and cascading enhancement nodes can extract more useful feature information. However, the corner position error of CE-BLS gradually increases with time, which indicates that cascading enhancement nodes alone lead to the accumulation of errors, and its effect tends to be poorer than that of I-BLS. The CFE-BLS exhibits superior tracking performance in comparison to I-BLS, which indicates that cascading feature nodes and cascading enhancement nodes can extract more valuable information from feature mapping. CFE-BLS has better tracking performance than CF-BLS and CE-BLS, and its angular position trajectory always stays near the demonstration trajectory with a small deviation, but does not converge to the demonstration trajectory. The angular position trajectory of the control strategy CFE-EN-BLS proposed in this paper always tracks the demonstration trajectory with a very small error and the deviation has a tendency to converge to zero. Besides, the angular velocity trajectory also converges to the demonstration velocity trajectory more smoothly compared with the other control strategies. From Fig. 3(b) and Fig. 3(d), it can be seen that the control strategy in this paper can converge quickly with minimum fluctuation. This illustrates the strong contribution of cascaded feature enhancement to the performance improvement of the control model, while it shows the superiority of elastic net regression in preventing the redundancy of feature extraction.

B. Real-Wold Experiments

The Aubo-i5 robotic arm experimental platform used is shown in Fig. 4. The trained control strategy is used to control the Aubo-i5 robotic arm job operation. The proposed approach calculates the upper-level control instructions, whereas the system itself furnishes the underlying execution mechanism. The designed robotic arm operation task is shown in Fig. 5(a) with a running time of 190 seconds. The actual tracking trajectory of the robotic arm is shown in Fig. 5(b).

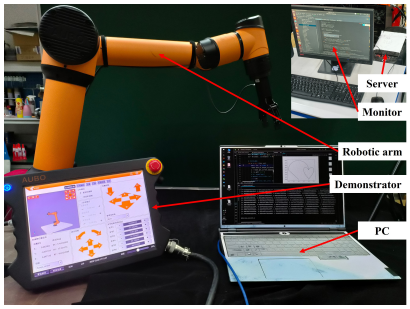
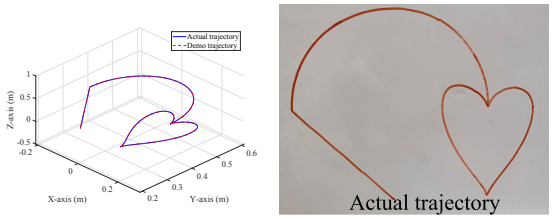


Fig. 4. Aubo-i5 robotic arm experiment platform. It mainly includes the robotic arm, personal computer (PC), server, demonstrator, and monitor.

Snapshots of the physical experiments based on the motion control of the Aubo-i5 robotic arm are shown in Fig. 6. To clearly illustrate the control efficacy,



(a) (b)

Fig. 5. Demonstration tracking trajectory designed (a) and actual trajectory (b) of the robotic arm. The first is linear motion, the next is circular motion, and the final is curved motion.

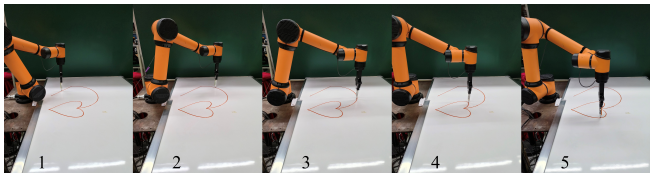
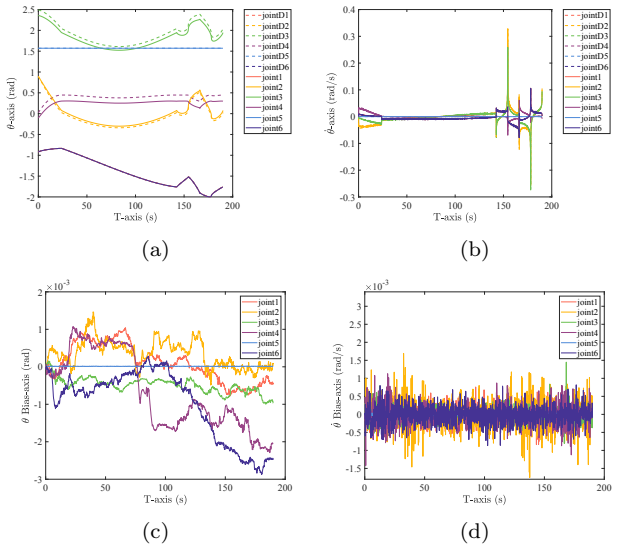


Fig. 6. Tracking trajectory of robotic arm movement.

we present the desired and actual angles, angular velocities, and their respective errors for each joint of the robotic arm, and the outcomes are displayed in Fig. 7. The proposed control strategy can maintain good performance throughout the operation. However, there are still some deviations in the angular position tracking process. There is a small oscillation in the



(a) (b)

(c) (d)

Fig. 7. Experimental results on Aubo-i5 robotic arm motion control. (a) and (b) show the output of the rotation angle and angular velocity of each joint, with the dashed line representing the expected value and the solid line representing the actual value. (c) and (d) show the angular position error and angular velocity error of each joint.

angular velocity tracking, as shown in Fig. 7(d). The algorithm in this paper converges to the neighborhood of 0 with a smaller position deviation than the simulation experiments, indicating that the proposed algorithm can be better applied to the robotic arms motion control. The small oscillations for the angular position and velocity deviations are because there may be a certain degree of interference in the low-level actuation system of the joints. From the experimental results, it can be seen that the actual rotation angle of the control strategy proposed in this paper always converges to the neighborhood of the desired value or with a small error. The effectiveness of the proposed control strategy is verified.

V. CONCLUSION

This paper proposes a robotic arm motion control strategy based on CFE-EN-BLS. Firstly, the motion control strategy of a robotic arm based on CFE-BLS is constructed. Secondly, this paper designs the intelligent control strategy of a robotic arm in combination with ElasticNet regression. Finally, the prerequisites for the stability of the control method are analyzed in combination with the Lyapunov theory. Both simulation studies and practical experiments verify that the introduced control strategy is superior to the other compared control models in terms of tracking accuracy, stability, and convergence performance.

References

[1] S. Hayakawa, W. Wan, K. Koyama, and K. Harada, "A dual-arm robot that manipulates heavy plates with the support of a vacuum lifter," IEEE Transactions on Automation Science and Engineering, vol. 20, no. 4, pp. 2808–2821, 2023.

[2] Q. Yang, X. Du, Z. Wang, Z. Meng, Z. Ma, and Q. Zhang, "A review of core agricultural robot technologies for crop productions," Computers and Electronics in Agriculture, vol. 206, p. 107701, 2023.

[3] D. Meng, H. Xu, H. Xu, H. Sun, and B. Liang, "Trajectory tracking control for a cable-driven space manipulator using time-delay estimation and nonsingular terminal sliding mode," Control Engineering Practice, vol. 139, p. 105649, 2023.

[4] Ohuchida and Kenoki, "Robotic surgery in gastrointestinal surgery," Cyborg and Bionic Systems, 2020.

[5] H. Jana and van Doorn Jenny, "Robots do not judge: service robots can alleviate embarrassment in service encounters." Journal of the Academy of Marketing Science, vol. 51, pp. 11–18, 2022.

[6] X. Lu, X. Zhang, G. Zhang, J. Fan, and S. Jia, "Neural network adaptive sliding mode control for omnidirectional vehicle with uncertainties," ISA Transactions, vol. 86, pp. 201–214, MAR 2019.

[7] H. He, C.-l. Lu, Y. Wen, G. Saunders, P. Yang, J. Schoonover, J. Wason, A. Julius, and J. T. Wen, "High-speed high-accuracy spatial curve tracking using motion primitives in industrial robots," in 2023 IEEE INTERNATIONAL CONFERENCE ON ROBOTICS AND AUTOMATION (ICRA 2023), 2023, pp. 12 289–12 295.

[8] Y. He, X. Li, Z. Xu, X. Zhou, and S. Li, "Collaboration of multiple scara robots with guaranteed safety using recurrent neural networks," Neurocomputing, vol. 456, pp. 1–10, 2021.

[9] N. Tan, P. Yu, Z. Zhong, and Y. Zhang, "Data-driven control for continuum robots based on discrete zeroing neural networks," IEEE Transactions on Industrial Informatics, vol. 19, no. 5, pp. 7088–7098, 2023.

[10] W. Zhang, Z. Ling, S. Heinrich, X. Ding, and Y. Feng, "Walking speed learning and generalization using seq2seq gated and adaptive continuous-time recurrent neural network (s2s-gactrnn) for a hip exoskeleton," IEEE-ASME Transactions on Mechatronics, vol. 28, no. 4, pp. 2375–2386, 2023.

[11] S. Balula, D. Liao-McPherson, A. Rupenyan, and J. Lygeros, "Data-driven reference trajectory optimization for precision motion systems," Control Engineering Practice, vol. 144, 2024.

[12] D. Papageorgiou, G. t. Siguroardottir, E. Falotico, and S. Tolu, "Sliding-mode control of a soft robot based on data-driven sparse identification," Control Engineering Practice, vol. 144, 2024.

[13] J. Duan, Y. Ou, J. Hu, Z. Wang, S. Jin, and C. Xu, "Fast and stable learning of dynamical systems based on extreme learning machine," IEEE Transactions on Systems Man Cybernetics-Systems, vol. 49, no. 6, pp. 1175–1185, 2019.

[14] G. Zuo, J. Zhou, L. Liu, and D. Gong, "Mpmc-frame: Multiplatform migration control framework for manipulator control," Control Engineering Practice, vol. 145, 2024.

[15] Z. Yan, X. Lai, Q. Meng, M. Wu, J. She, and M. Iwasaki, "Modeling, analysis, and adaptive neural modified-backstepping control of an uncertain horizontal pendubot with double flexible joints," Control Engineering Practice, vol. 139, 2023.

[16] C. Relano, J. Munoz, and C. A. Monje, "Gaussian process regression for forward and inverse kinematics of a soft robotic arm," Engineering Applications of Artificial Intelligence, vol. 126, no. D, 2023.

[17] M. Yoon, M. Kang, D. Park, and S.-E. Yoon, "Learning-based initialization of trajectory optimization for path-following problems of redundant manipulators," in 2023 IEEE INTERNATIONAL CONFERENCE ON ROBOTICS AND AUTOMATION (ICRA 2023). IEEE; IEEE Robot & Automat Soc, 2023, pp. 9686–9692.

[18] A. Ayyad, M. Halwani, D. Swart, R. Muthusamy, F. Almaskari, and Y. Zweiri, "Neuromorphic vision based control for the precise positioning of robotic drilling systems," Robotics and Computer-Integrated Manufacturing, vol. 79, 2023.

[19] H.-T. Nguyen, C. C. Cheah, and K.-A. Toh, "An analytic layer-wise deep learning framework with applications to robotics," Automatica, vol. 135, 2022.

[20] C. L. P. Chen and Z. Liu, "Broad learning system: An effective and efficient incremental learning system without the need for deep architecture," IEEE Transactions on Neural Networks and Learning Systems, vol. 29, no. 1, pp. 10–24, 2018.

[21] S. Xu, T. Xu, D. Li, C. Yang, C. Huang, and X. Wu, "A robot motion learning method using broad learning system verified by small-scale fish-like robot," IEEE Transactions on Cybernetics, vol. 53, no. 9, pp. 6053–6065, 2023.

[22] C.-F. Hsu, B.-R. Chen, and B.-F. Wu, "Fuzzy broad learning adaptive control for voice coil motor drivers," International Journal of Fuzzy Systems, vol. 24, no. 3, pp. 1696–1707, 2022.

[23] H. Huang, T. Zhang, C. Yang, and C. L. P. Chen, "Motor learning and generalization using broad learning adaptive neural control," IEEE Transactions on Industrial Electronics, vol. 67, no. 10, pp. 8608–8617, 2020.

[24] H. Huang, C. Yang, and C. L. P. Chen, "Optimal robot-environment interaction under broad fuzzy neural adaptive control," IEEE Transactions on Cybernetics, vol. 51, no. 7, pp. 3824–3835, 2021.

[25] S. Xu, J. Liu, C. Yang, X. Wu, and T. Xu, "A learning-based stable servo control strategy using broad learning system applied for microrobotic control," IEEE Transactions on Cybernetics, vol. 52, no. 12, pp. 13 727–13 737, 2022.

[26] G. Zuo, J. Zhou, D. Gong, and G. Huang, "Intelligent servo control strategy for robot joints with incremental bayesian fuzzy broad learning system," IEEE-ASME Transactions on Mechatronics, vol. 28, no. 4, pp. 2029–2037, 2023.

[27] M. Mou and X. Zhao, "Gated broad learning system based on deep cascaded for soft sensor modeling of industrial process," IEEE Transactions on Instrumentation and Measurement, vol. 71, 2022.