scientific reports

Check for updates

OPEN

Adaptive robust control of tea-picking-manipulator's position tracking based on dead zone compensation with modified RBF

Yu Han[1,2,3]✉, Zhiyu Song[3]✉, Wenyu Yi[4]✉ & Caixue Zhan[3,5]

Neural Network has been used in approximation of dead zone nonlinearity when modeling the manipulator control systems. However the existed method fail to minimize the possible input saturation effect and the NN mapping accuracy also be degraded, which leads to degrading in tracking precise and accuracy. To establish an accurate control model of the tea picking robot, an adaptive compensator of m-RBF (modified radial basis function neural network) and adaptive law were designed aimed at the nonlinearity units and dead zone. The closed-loop tracking error of the proposed control system is eventually going to be stable and bounded. A simulation was carried out with Simulink, which show that m-RBF provides a good approximation to characteristic of Dead zone nonlinearity, and that the control scheme based on m-RBF had a excellent and stable tracking accuracy. The tea picking experiment with six-axis manipulator verifies the effectiveness of the proposed algorithm, in which the proposed method got a higher score of 95.3, near two times that of traditional PID control methods. It can be drown that the m-RBF has a faster learning rate and can avoid local minima; the control scheme based on m-RBF has a excellent performance on control accuracy, robustness and self-adaption, which is especially appropriate for real-time control, like tea picking robot.

Keywords Dead zone nonlinearity, RBF neural network, Manipulator, Tracking, Control, Tea harvesting

Tea, the Chinese national drink, is the carrier of Chinese culture and an important cash crop in China with an annual output value of more than 150 billion yuan; among them, the output value of famous tea accounts for more than 50%[1]. Because of the strict requirements on the appearance of famous tea, it still needs a lot of manual labor to pick it. Due to the shortage of agricultural labor force, the cost of picking high-quality tea has exceeded 60% of the production cost[2]. However, due to the strong seasonal nature of tea, it will cause huge economic losses (due to the decline in quality) when not picked: the development of tea industry is severely restricted. The realization of famous tea picking machine replacement is imminent.

Both domestic and foreign tea picking machinery adopt the integral picking principle of "reciprocating cutting", which has no selectivity and the buds and leaves are seriously damaged, so it is not applicable for picking famous tea (high quality tea). Due to the unique Chinese characteristics of famous tea, only a small number of studies on tea picking technology have been carried out abroad. Based on feature extraction and object detection, Vu et al.[3] studied the automatic counting method of tea shoots based on machine vision and realized regional detection of tea leaves. Thangavel et al.[4] adopted key frame extraction method to identify tea leaves and realized leaf region detection. Karunasena et al.[5] used machine learning and image processing techniques to identify tea bud and leaf. Jain et al.[6] designed a picking robot based on double R manipulator, which was equipped with small scissors at the end to cut tea shoot. Combined with the Internet of Things technology, the scheduling control technology of the picking robot was studied. Motokura et al.[7] conducted a preliminary study on the pose and trajectory planning of the end-effector of tea picking robot. Due to less research, there is no mature and available intelligent tea picking technology abroad.

[1]School of Automation, Southeast University, Nanjing 210096, China. [2]School of Mathematics, Southeast University, Nanjing 210096, China. [3]Ministry of Agriculture and Rural Affairs, Nanjing Institute of Agricultural Machanization, Nanjing 210014, China. [4]Key Laboratory of Agricultural Equipment Technology for Hilly and Mountainous Areas, Ministry of Agriculture and Rural Affairs, Chengdu 610066, China. [5]The Key Laboratory for Crop Production and Smart Agriculture of Yunnan Province, Kunming 650540, China. ✉email: hanyu@caas.cn; songzy1984@163.com; ywycd@scsaas.cn

The research on the picking technology of famous tea mainly focuses on the picking robot technology based on computer vision. In terms of tea bud visual recognition, picture-processed-based methods have been well investigated, such as Yang et al.[8], Jin et al.[9], Wang et al.[10] and Qi et al.[11]. These methods have poor generalization ability for they are relied heavily on artificial target's features that vary from different tea trees, ripeness and weather conditions. To kill the problem, deep learning began to be applied to tea bud identification, such as Lv et al.[12] adopted Alex-Net CNN (Convolutional Neural Network), Yang et al.[13,14], Li et al.[15] and Gong et al.[16] adopted YOLO (you look only once), Chen et al.[17] adopted deep CNN to identify the tea bud target. Their In-field experiment showed a picking success rate of 65%. Due to the diverse tea varieties, interleaved and dense distribution and complex background, the problem of accurate identification and positioning of tea buds has not been effectively solved.

Refers to picking robot ontology, Chen[18] carried out the research and design of the parallel mechanism and end-effector of tea picking robot, and developed the prototype. Wei[19] and Sun[20] have also carried out researches on tender shoot identification, picking point determination, parallel mechanism, bionic picking finger, robot ontology, intelligent control, etc., and achieved a series of breakthrough results. However, both the picking efficiency and quality of the robot are still relatively low. In the transmission system of picking robot, the main factors affecting the control performance of the system are the existence of a wide range of nonlinearities in the process of power transmission, such as backlash, dead zone, friction and saturation, which become the main factors affecting the control accuracy[21]. In most cases, these nonlinear phenomena are unknown and time varying, and can have a great impact on the system. As one of the main nonlinear characteristics, dead zone nonlinear has always been one of the main research objects in the field of high precision transmission control. Dead zones have a great impact on the control system, the most common of which is to reduce the accuracy of the static output and make the dynamic output appear "flat top"[22]. Limit cycles or system instability may also result because the system does not respond within the dead zone. Consequently, it may cause the Tea-picking-manipulator's shaking and inaccuracy of tracking target, leading to damage to tea leaves and the fail of picking. Therefore, the compensation of dead zone becomes an important subject of high precision control system.

The mathematical description of a dead zone is a static nonlinear function used to describe the insensitivity of a system to small signals[23]. When the signal enters the dead zone, there will be considerable loss of signal information, which will cause the limit cycle and tracking error of the system. From a feedback perspective, dead zones represent a "loss of information". Like many other discontinuous nonlinear phenomena, the main cause of dead zone is friction. Due to the non-differentiable nonlinearity of the dead zone, it is very difficult to compensate the dead zone, which greatly increases the complexity of the drive system control[24]. It is of great theoretical significance and engineering value to study the dead zone nonlinearity in mechanical and electromechanical transmission systems. However, the control problem of robot with unmeasurable constrained states and other unknown nonlinearities is not properly addressed.

In early years, various control methods have been designed to handle deadzone, which enhanced the performance of control systems, such as backstepping control, model predictive control, variable structure control, etc.[25]. However, These methods work on the condition that some parameters of deadzone functions are known. In practice, these parameters are always hard to measure, especially when considering the effect of time. Deadzone function is inherently nonlinear. Wang et al.[26] focus on the adaptive control of nonlinear dynamic systems with an unknown deadzone. The fuzzy logic control methods have been employed generally to approximate the nonlinear systems. Tong et al.[27] investigated the problem of the adaptive fuzzy decentralized output-feedback control for a category of switched nonlinear strict-feedback large-scale systems. Pei et al.[28] proposed a robust deadzone compensation method against parameter variations based on kalman filter and neural networks. Because the neural network can approximate any nonlinear continuous function with any precision, it has the ability of self-adaptation and self-learning for complex uncertain problems. Lewis et al.[29] studied the neural network control of the dead zone link in the robot actuator, and adopted network to carry out adaptive compensation for the dead zone in the control system, which is a classic research achievement in neural network control. Zhou et al.[30] designed a NN(neural network) controller for the robotic manipulators subject to the dead zone. The NN approximation approach is employed to approximate the intermediate control signals with unknown nonlinear functions, reduced the number of estimated parameters. Lu et al.[31] used RBF to generate several networks to replace the BP network in the original compensator, and designed the RBF(Radial Basis Function) network adaptive robust control with dead zone compensation, which could not only greatly reduce the system parameters, but also make the network initialization more clear and definite. Bensidhoum et al.[32] used RBF neural network to approximate the unknown nonlinear functions in contrl system of robot manipulators with unknown input dead-zone. The clustering algorithm is used to determine the center of radial basis function in RBF neural network, but how to determine the appropriate number of clusters has not been well solved, which degrads the RBF mapping accuracy. Also, only the proportional action is used in the control scheme to avoid the measurement noise, which fails to minimize the possible input saturation effect (the input exceeds its limit), leading to degrading in tracking precise and accuracy.

To adress this problem in tea picking robot, this paper propose an control skeme based on modified RBF to approximate the deadzone of tea picking manupulator. The main work and contributions are as follow:

(1) Proposed a modified Radial Basis Function (m-RBF) neural network to compensate the dead zone in tea picking robot control system, in which a new clustering algorithm and clustering automatic termination criterion are introduced to improve RBF and the mapping accuracy.
(2) Construct the control system with m-RBF compensator, and design the adaptive law of neural network compensator. Also, Introduce the mathematical concepts of GL (General Linear) matrix and GL multiplication operator to prove the uniformly bounded stability of n joint robot system strictly.

(3) The control system with m-RBF compensator is simulated with Matlab, showing the superiority of the control scheme based on m-RBF compensation. And the tea picking experiment verifies the effectiveness of the control system, the tracking becoming rapid and smooth with less breakdown of chattering and being stuck.

The rest of this article is organized as follows. In Section "Tea picking manipulator model", tea picking manipulator model is presented. The adaptive control based on m-RBF neural network and its convergence analysis are given in Section "Model". To show the effectiveness of the proposed control scheme, an simulation example and the picking experiment are conducted in Section "Simulation and experiment". Finally, conclusions and Discussion are drawn in Section "Conclusion and discussion".

Tea picking manipulator model

Here, we write the dynamic equation of the N-joint manipulator as:

$$M\left(q\right)\ddot{q}+\ddot{V}_{m}\left(q,\dot{q}\right)\dot{q}+G\left(q\right)=\tau \tag{1}$$

where Type of $q(t)\in R^{n}$ for joint angular displacement quantity, $M(q)\in R^{n}$ as the inertia matrix, $v(q,\dot{q})\in R^{n}$ as centrifugal force and Coriolis force, $G(q)\in R^{n}$ as gravity, $\tau\in R^{n}$ for control of torque.

The mathematical expression of the dead zone is a static nonlinear function that describes the insensitivity of the system to small signals, as shown in Fig. 1. In most closed-loop systems, the dead zone has a bad effect on both transient and steady-state performance. The signal entering the dead zone is accompanied by information loss, which easily leads to the limit cycle and large tracking error.

Generally speaking, the dead zone phenomenon in the motion control system can be expressed as:

$$\tau=D\left(u\right)=\begin{cases} g\left(u\right)<0 & u\leq d_{-} \\ 0 & d_{-}<u<d_{+} \\ h\left(u\right)>0 & u\geq d_{+} \end{cases} \tag{2}$$

where u is the input signal, h and g are two smooth nonlinear functions, and τ is the output of the dead zone.

The description of formula (2) covers various possible situations in the dead zone phenomenon. $h(u)$,$g(u)$,d_{+} and d- are all unknown. Suppose $h(u)$ and $g(u)$ are monotonically invertible increasing functions on the definition interval, so there is an inverse $D^{-}(w)$ with a dead zone:

$$D^{-1}\left(w\right)=\begin{cases} g^{-}\left(w\right)<0 & u<0 \\ 0 & u=0 \\ h^{-}\left(w\right)>0 & u>0 \end{cases} \tag{3}$$

Then we have

$$D(D^{-1}\left(w\right))=w \tag{4}$$

In order to compensate the negative effects of the unknown dead zone, the neural network compensator is put in front of the control input to compensate the nonlinear link of the dead zone.

The structure of robot moment arm control system is shown in Fig. 2. The control objective is to expect the output $\boldsymbol{w}$ of the controller to be the same as the output $\boldsymbol{\tau}$ of the actuator. Therefore, RBF neural network can be used to compensate. In Fig. 2, $\boldsymbol{w}$ is the ideal control input, $\boldsymbol{u}$ is the actual control input of the robot after dead zone compensation, and $\boldsymbol{\tau}$ is the control signal after dead zone compensation.

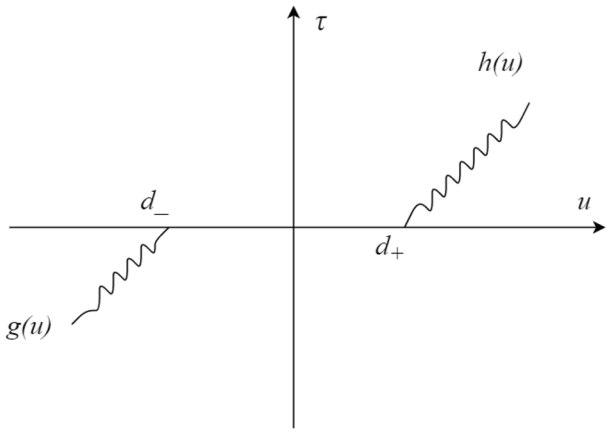

Fig. 1. Characters of dead zone nonlinearity.

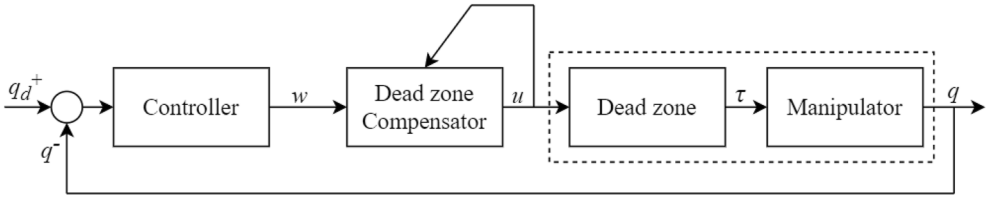

Fig. 2. Control system based on neural network compensation.

Model

This section will setup the neural network model to establish the compensator model for the dead zone. Also, the control system stability will be analyzed.

GL matrix and GL multiplication operator

Define symbol "{·}" as GL matrix and "·" as its multiplication operator. $\boldsymbol{I_0}$ as an integer set, $\theta_{ij}, \xi_{ij} \in R_{ij}^n$, where $n_{ij} \in I_0, i = 1, \ldots, n, j = 1, \ldots, n$. For function approximation, θ_{ij} can be regarded as a weight vector and ξ_{ij} can be regarded as a Gaussian basis function vector. The GL vector $\{\theta_k\}$ and its transpose, $\{\theta_k\}^T$, are defined as follows:

$$\{\theta_k\} = \{\theta_{k1}\theta_{k2}\ldots\theta_{kn}\} \tag{5}$$

$$\{\theta_k\}^T = \left\{\theta_{k1}{}^T\theta_{k2}{}^T\ldots\theta_{kn}{}^T\right\} \tag{6}$$

The GL Matrix $\{\theta_k\}$ and its transpose $\{\theta_k\}^T$ are defined as follows:

$$\{\theta\} = \begin{Bmatrix} \{\theta_1\} \\ \{\theta_2\} \\ \vdots \\ \{\theta_n\} \end{Bmatrix}, \{\theta\}^T = \begin{Bmatrix} \{\theta_1\}^T \\ \{\theta_2\}^T \\ \vdots \\ \{\theta_n\}^T \end{Bmatrix}$$

For a given GL matrix, $\{\xi\} = \begin{Bmatrix} \xi_{11} & \xi_{12} & \cdots & \xi_{1n} \\ \xi_{21} & \xi_{22} & \cdots & \xi_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ \xi_{n1} & \xi_{n2} & \cdots & \xi_{nn} \end{Bmatrix} = \begin{Bmatrix} \{\xi_1\} \\ \{\xi_2\} \\ \vdots \\ \{\xi_n\} \end{Bmatrix}$,

GL Multiplication of $\{\theta\}^T$ and $\{\xi\}$ is defined as an n by n matrix:

$$[\{\theta\}^T \cdot \{\xi\}] = \begin{bmatrix} \theta_{11}^T\xi_{11} & \theta_{12}^T\xi_{12} & \cdots & \theta_{1n}^T\xi_{1n} \\ \theta_{21}^T\ \xi_{21} & \theta_{22}^T\xi_{22} & \ldots & \theta_{2n}^T\xi_{2n} \\ & \vdots\ \vdots & \ddots & \vdots \\ \theta_{n1}^T\ \xi_{n1} & \theta_{n2}^T\xi_{n2} & \ldots & \theta_{nn}^T\xi_{nn} \end{bmatrix} \tag{7}$$

Let $\boldsymbol{\Gamma}_k = \boldsymbol{\Gamma}_k^T = [\gamma_{k1}\gamma_{k2}\ldots\gamma_{kn}]$, $\gamma_{k2} \in R^{m\times n_j}$, $m = \sum_{j=1}^n n_j$, Then the GL multiplication of a square matrix and a GL vector is defined as

$$\boldsymbol{\Gamma}_k \cdot \{\xi_k\} = \{\Gamma_k\} \cdot \{\xi_k\} = \{\gamma_{k1}\xi_{k1}\gamma_{k2}\xi_{k2}\ldots\gamma_{kn}\xi_{kn}\} \in R^{n\times m} \tag{8}$$

In the mixed calculation of matrix and GL matrix with respect to GL multiplication operator, the GL multiplication is computed first. For example: $\{\boldsymbol{A}\}\cdot\{\boldsymbol{B}\}\boldsymbol{C}$, first calculate $\{\boldsymbol{A}\}\cdot\{\boldsymbol{B}\}$, and then multiply the result with $\boldsymbol{C}$. For further information about GL matrix and operator, please refer to the work of Ge[33].

Neural network model

RBF neural network

The neural network can approximate any nonlinear continuous function with any precision and has adaptive and self-learning ability for complex uncertain problems. A Radial Basis Function (RBF) neural network is a three-layer feed-forward network with a single hidden layer. The function of RBF network is a Gaussian basis function, and its value is non-zero in the finite range of input space, so RBF network is a local approximation neural network. The mapping from the input layer to the output layer is nonlinear, while the mapping from the hidden layer space to the output layer space is linear. Due to its local approximation property, it can greatly accelerate the learning speed and avoid the local minimum problem, which is suitable for the application requirements of

real-time control. The neural network control scheme composed of RBF network can effectively improve the control precision, robustness and adaptability of the system.

FCM (Fuzzy C-Means) clustering algorithm is often used to determine the center of radial basis function in RBF neural network, but how to determine the appropriate number of clusters has not been well solved. In order to obtain reasonable central parameters of radial basis function, a new clustering algorithm and clustering automatic termination criterion are introduced here to improve RBF and improve the mapping accuracy.

m-RBF neural network

Here, the RBF neural network is improved by two stages: (1) the central parameter of radial basis function is learned by using the Reduced Clustering algorithm[34], and the number of RBF basis functions is determined by applying the self-terminating clustering criterion; (2) The output weight is adjusted by gradient descent method. The self-terminating clustering criterion in the first stage is that: If the ratio between the current maximum density value and the initial maximum density value is less than some very small given value, the clustering is terminated. That is, when the current cluster center contains very few data points, you can ignore the cluster center and end the cluster.

The specific learning steps are as follows:

Step 1: the density index of each data is calculated according to formula (9) and saved in the set *A*. The data point with the highest density index is selected as the first clustering center.

$$D_i = \sum_{j=1}^{p} exp\left[-\frac{\|x_i - x_j\|^2}{\left(\gamma_a/2\right)^2}\right] \tag{9}$$

where γ_a define the neighborhood radius of each data. x_i, x_j denote the *i* th and *j* th date of the N-dimensional data space.

Step 2: The *m* th cluster center is determined, the density index of each data is modified by formula (10), and the highest density index is searched.

$$D_i = D_i - D_{c_m} \sum_{j=1}^{p} exp\left[-\frac{\|x_i - x_j\|^2}{\left(\gamma_b/2\right)^2}\right] \tag{10}$$

where γ_b define a neighborhood with a significantly smaller density index. Generally, $\gamma_b > \gamma_a$, which avoid cluster centers that are very close together.

Step 3: Substitute the highest density index D_{max} into the criterion. If the criteria are met, the criteria are terminated and go to Step 4. If the criterion is not met, the point x_{c_m} is accepted as the *m*-th cluster center, and it is kept in the set *A*; updated: $D_{max} = D_{c_m}$, $m = m + 1$; then go to the second step.

Step 4: end of cluster.

Apparently, a total of *m*-1 clustering centers were identified. After the clustering center is determined, the width parameter can be determined by taking the nearest neighbor average distance to the neighboring points of each cluster center. By these steps we have the m-RBF.

Dead zone compensator design with m-RBF

The adaptive neural network compensator consists of two m-RBF neural networks, one m-RBF is used to estimate the nonlinear link in the actuator, and the other m-RBF is used to compensate the dead zone of the feedforward channel of the system. The basic structure of the neural network compensator is shown in Fig. 3.

where *u* is as follow:

$$u = w + \widehat{w}_{NN}(w) \tag{11}$$

The inverse of the dead-zone link can be expressed in the following equivalent form:

$$D^{-1}(w) = w + w_{NN} \tag{12}$$

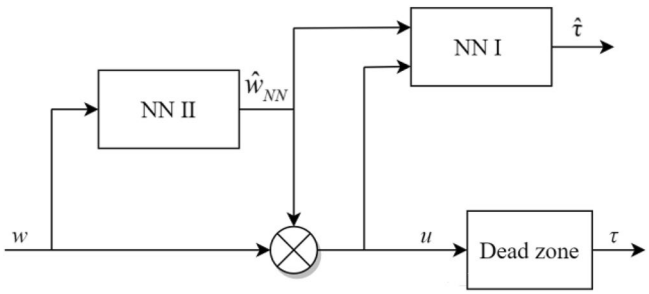

Fig. 3. *m*-RBF adaptive neural network compensator.

where w_{NN} can be wrote as follow,

$$\mathrm{w_{NN}} = \begin{cases} \mathrm{g}^{-1}(\mathrm{w}) - \mathrm{w} & \mathrm{w} < 0 \\ 0 & \mathrm{w} = 0 \\ \mathrm{h}^{-1}(\mathrm{w}) - \mathrm{w} & \mathrm{w} > 0 \end{cases} \tag{13}$$

According to Universal approximation of m-RBF neural network, we have,

$$\boldsymbol{D}(\boldsymbol{u}) = \{\boldsymbol{W}\}^{T} \cdot \{\boldsymbol{\sigma}(\boldsymbol{u})\} + \boldsymbol{\varepsilon}(\boldsymbol{u}) \tag{14}$$

$$\mathbf{w_{NN}} = \{\mathbf{W_i}\}^{\mathbf{T}} \cdot \{\boldsymbol{\sigma}_\mathbf{i}(\mathbf{u})\} + \boldsymbol{\varepsilon}_\mathbf{i}(\mathbf{u}) \tag{15}$$

Where $\boldsymbol{\varepsilon}(\boldsymbol{u})$ and $\boldsymbol{\varepsilon_i}(\boldsymbol{u})$ is the modeling error of neural network, $\boldsymbol{W}$ and $\boldsymbol{W}_i$ is the ideal weight, $\boldsymbol{\sigma}$ and $\boldsymbol{\sigma}_i$ is the output of the radial basis function.

Let $\widehat{\boldsymbol{W}}$ and $\widehat{\boldsymbol{W}}_i$ refer to ideal estimation of RBF weights. Refered to Wang's work[35], we can estimate the corresponding links through two RBF neural networks:

$$\widehat{\mathbf{D}}(\mathbf{u}) = \{\widehat{\mathbf{W}}\}^{\mathbf{T}} \cdot \{\boldsymbol{\sigma}(\mathbf{u})\} \tag{16}$$

$$\widehat{\mathbf{w}}_{\mathbf{NN}} = \{\widehat{\mathbf{W}}_\mathbf{i}\}^{\mathbf{T}} \cdot \{\boldsymbol{\sigma}_\mathbf{i}(\mathbf{u})\} \tag{17}$$

Define $\widetilde{W} = W - \widehat{W}, \widetilde{W}_i = W_i - \widehat{W}_i$ as the estimation error of RBF weights.

It can be seen from Fig. 2 that the purpose of the design of the control system is to make the output $\boldsymbol{\tau}$ of the controller and the robot input $\boldsymbol{w}$ after the dead zone tend to be as consistent as possible. The mathematical relationship between the two and the corresponding proof are given below.

In Fig. 2, given the m-RBF neural network compensator $\widehat{w}_{NN}$, the mathematical relation between the control input $\boldsymbol{w}$ and the control output $\boldsymbol{\tau}$ in the dead zone is as follows:

$$\boldsymbol{\tau} = \mathbf{w} - \left[\left\{\widehat{\mathbf{W}}\right\}^{\mathbf{T}} \cdot \left\{\boldsymbol{\sigma}'(\mathbf{u})\right\}\right] \cdot \left[\left\{\widetilde{\mathbf{W}}_\mathbf{i}\right\}^{\mathbf{T}} \cdot \{\boldsymbol{\sigma}_\mathbf{i}(\mathbf{w})\}\right] + \left[\left\{\widetilde{\mathbf{W}}\right\}^{\mathbf{T}} \cdot \left\{\boldsymbol{\sigma}'(\mathbf{u})\right\}\right] \cdot \left[\left\{\widehat{\mathbf{W}}_\mathbf{i}\right\}^{\mathbf{T}} \cdot \{\boldsymbol{\sigma}_\mathbf{i}(\mathbf{w})\}\right] + \mathbf{d}(\mathbf{t}) \tag{18}$$

The non-matching item of the model is:

$$\mathbf{d}(\mathbf{t}) = -\left[\left\{\widetilde{\mathbf{W}}\right\}^{\mathbf{T}} \cdot \left\{\boldsymbol{\sigma}'(\mathbf{u})\right\}\right] \cdot \left[\{\mathbf{W_i}\}^{\mathbf{T}} \cdot \{\boldsymbol{\sigma}_\mathbf{i}(\mathbf{w})\}\right] - \mathbf{b}(\mathbf{t}) + \boldsymbol{\varepsilon}(\mathbf{u}) \tag{19}$$

$$\mathbf{b}(\mathbf{t}) = \left\{\widetilde{\mathbf{W}}\right\}^{\mathbf{T}} \cdot \left\{\left[\boldsymbol{\sigma}'\left(\mathbf{w} + \left\{\widehat{\mathbf{W}}_\mathbf{i}\right\}^{\mathbf{T}} \cdot \{\boldsymbol{\sigma}_\mathbf{i}(\mathbf{w})\}\right)\right] \cdot [\boldsymbol{\varepsilon}_\mathbf{i}(\mathbf{w})]\right\} + \{\mathbf{W}\}^{\mathbf{T}} \cdot \left\{\mathbf{R}\left(\widetilde{\mathbf{W}}_\mathbf{i}, \mathbf{w}\right)\right\} + \boldsymbol{\varepsilon}(\mathbf{w} + \mathbf{w_{NN}}) \tag{20}$$

The proof of mathematical relationship between w and τ

According to Eqs. (2), (11) and (14), the dead zone output of the k^{th} joint in the N-joint robot can be written:

$$\boldsymbol{\tau}_\mathbf{k} = \mathbf{D}(\mathbf{u_k}) = \mathbf{W_k^T} \cdot \boldsymbol{\sigma}(\mathbf{u_k}) + \boldsymbol{\varepsilon}(\mathbf{u_k}) = \mathbf{W_k^T} \cdot \boldsymbol{\sigma}(\mathbf{w_k} + \widehat{\mathbf{w}}_{\mathbf{NNk}}) + \boldsymbol{\varepsilon}(\mathbf{w_k} + \widehat{\mathbf{w}}_{\mathbf{NNk}}) \tag{21}$$

From Eq. (4) and (12), yields

$$\mathbf{w_k} = \mathbf{D}\left(\mathbf{D}^{-1}(\mathbf{w_k}\right) = \mathbf{D}(\mathbf{w_k} + \mathbf{w_{NNk}}) \tag{22}$$

By substituting Eqs. (21) and (15) into Eqs. (22), we have:

$$\begin{aligned} \boldsymbol{w_k} &= \boldsymbol{W_k^T} \cdot \boldsymbol{\sigma}(\boldsymbol{w_k} + \boldsymbol{w_{NNk}}) + \boldsymbol{\varepsilon}(\boldsymbol{w_k} + \boldsymbol{w_{NNk}}) \\ &= \mathbf{W_k^T} \cdot \boldsymbol{\sigma}\left(\mathbf{w_k} + \mathbf{W_{ik}}^{\mathbf{T}} \cdot \boldsymbol{\sigma}_\mathbf{i}(\mathbf{w_k}) + \boldsymbol{\varepsilon}_\mathbf{i}(\mathbf{w_k})\right) + \boldsymbol{\varepsilon}(\mathbf{w_k} + \mathbf{w_{NNk}}) \\ &= \mathbf{W_k^T} \cdot \boldsymbol{\sigma}\left(\mathbf{w_k} + \widehat{\mathbf{W}}_{\mathbf{ik}}^{\mathbf{T}} \cdot \boldsymbol{\sigma}_\mathbf{i}(\mathbf{w_k}) + \widetilde{\mathbf{W}}_{\mathbf{ik}}^{\mathbf{T}} \cdot \boldsymbol{\sigma}_\mathbf{i}(\mathbf{w_k}) + \boldsymbol{\varepsilon}_\mathbf{i}(\mathbf{w_k})\right) + \boldsymbol{\varepsilon}(\mathbf{w_k} + \mathbf{w_{NNk}}) \end{aligned}$$

Let $\boldsymbol{x} = \boldsymbol{w_k} + \widehat{\boldsymbol{W}}_{ik}^{T} \cdot \boldsymbol{\sigma}_i(\boldsymbol{w_k}) + \widetilde{\boldsymbol{W}}_{ik}^{T} \cdot \boldsymbol{\sigma}_i(\boldsymbol{w_k}) + \boldsymbol{\varepsilon}_i(\boldsymbol{w_k})$

Then $\boldsymbol{x}_0 = \boldsymbol{w_k} + \widehat{\boldsymbol{W}}_{ik}^{T} \cdot \boldsymbol{\sigma}_i(\boldsymbol{w_k})$

And $\boldsymbol{x} - \boldsymbol{x} = \widetilde{\boldsymbol{W}}_{ik}^{T} \cdot \boldsymbol{\sigma}_i(\boldsymbol{w_k}) + \boldsymbol{\varepsilon}_i(\boldsymbol{w_k})$

Let $f(x) = \sigma(x)$, According to the first order Taylor expansion formula, yields

$$\boldsymbol{R}(\boldsymbol{\xi}) = \boldsymbol{R}\left(\widetilde{\boldsymbol{W}}_{ik}, \boldsymbol{w_k}\right) = \frac{1}{2!}\boldsymbol{\sigma}''(\boldsymbol{\xi}_0)\left(\widetilde{\boldsymbol{W}}_{ik}^{T} \cdot \boldsymbol{\sigma}_i(\boldsymbol{w_k}) + \boldsymbol{\varepsilon}_i(\boldsymbol{w_k})\right)^2 \tag{23}$$

$$\boldsymbol{w_k} = \boldsymbol{W_k^T}(\boldsymbol{\sigma}(\boldsymbol{x}_0) + \boldsymbol{\sigma}'(\boldsymbol{x}_0)(\boldsymbol{x} - \boldsymbol{x}_0) + \boldsymbol{R}(\boldsymbol{\xi})) + \boldsymbol{\varepsilon}(\boldsymbol{w_k} + \boldsymbol{w_{NNk}})$$

$$= \boldsymbol{W}_{\boldsymbol{k}}^{\boldsymbol{T}}\boldsymbol{\sigma}\left(\boldsymbol{x}_0\right) + \boldsymbol{W}_{\boldsymbol{k}}^{\boldsymbol{T}}\boldsymbol{\sigma}^{'}\left(\boldsymbol{x}_0\right)\left(\boldsymbol{x}-\boldsymbol{x}_0\right) + \boldsymbol{W}_{\boldsymbol{k}}^{\boldsymbol{T}}\boldsymbol{R}(\boldsymbol{\xi}) + \varepsilon\left(\boldsymbol{w}_{\boldsymbol{k}} + \boldsymbol{w}_{\boldsymbol{NNk}}\right)$$

$$= \boldsymbol{W}_{\boldsymbol{k}}^{\boldsymbol{T}}\cdot\boldsymbol{\sigma}\left(\boldsymbol{w}_{\boldsymbol{k}} + \widehat{\boldsymbol{W}}_{\boldsymbol{ik}}^{\boldsymbol{T}}\cdot\boldsymbol{\sigma}_{\boldsymbol{i}}\left(\boldsymbol{w}_{\boldsymbol{k}}\right)\right) + \boldsymbol{W}_{\boldsymbol{k}}^{\boldsymbol{T}}\cdot\boldsymbol{\sigma}^{'}\left(\boldsymbol{w}_{\boldsymbol{k}} + \widehat{\boldsymbol{W}}_{\boldsymbol{ik}}^{\boldsymbol{T}}\cdot\boldsymbol{\sigma}_{\boldsymbol{i}}\left(\boldsymbol{w}_{\boldsymbol{k}}\right)\right)\left(\widetilde{\boldsymbol{W}}_{\boldsymbol{ik}}{}^{\boldsymbol{T}}\cdot\boldsymbol{\sigma}_{\boldsymbol{i}}\left(\boldsymbol{w}_{\boldsymbol{k}}\right) + \varepsilon_{\boldsymbol{i}}\left(\boldsymbol{w}_{\boldsymbol{k}}\right)\right) + \boldsymbol{W}_{\boldsymbol{k}}^{\boldsymbol{T}}\boldsymbol{R}(\boldsymbol{\xi}) + \varepsilon\left(\boldsymbol{w}_{\boldsymbol{k}} + \boldsymbol{w}_{\boldsymbol{NNk}}\right)$$

By substituting ***b(t)*** from Eq. (20) into the above equation, yields

$$\boldsymbol{w}_{\boldsymbol{k}} = \boldsymbol{W}_{\boldsymbol{k}}^{\boldsymbol{T}}\cdot\boldsymbol{\sigma}\left(\boldsymbol{w}_{\boldsymbol{k}} + \widehat{\boldsymbol{W}}_{\boldsymbol{ik}}^{\boldsymbol{T}}\cdot\boldsymbol{\sigma}_{\boldsymbol{i}}\left(\boldsymbol{w}_{\boldsymbol{k}}\right)\right) + \boldsymbol{W}_{\boldsymbol{k}}^{\boldsymbol{T}}\cdot\boldsymbol{\sigma}^{'}\left(\boldsymbol{w}_{\boldsymbol{k}} + \widehat{\boldsymbol{W}}_{\boldsymbol{ik}}^{\boldsymbol{T}}\cdot\boldsymbol{\sigma}_{\boldsymbol{i}}\left(\boldsymbol{w}_{\boldsymbol{k}}\right)\right)\widetilde{\boldsymbol{W}}_{\boldsymbol{ik}}^{\boldsymbol{T}}\cdot\boldsymbol{\sigma}_{\boldsymbol{i}}\left(\boldsymbol{w}_{\boldsymbol{k}}\right) + \boldsymbol{b}_{\boldsymbol{k}}(\boldsymbol{t}) \quad (24)$$

By combining Eqs. (11) and (17), Eq. (24) can be written as:

$$-\widetilde{\boldsymbol{W}}_{\boldsymbol{k}}{}^{\boldsymbol{T}}\cdot\boldsymbol{\sigma}^{'}\left(\boldsymbol{u}_{\boldsymbol{k}}\right)\boldsymbol{W}_{\boldsymbol{ik}}{}^{\boldsymbol{T}}\boldsymbol{\sigma}_{\boldsymbol{i}}\left(\boldsymbol{w}_{\boldsymbol{k}}\right) - \boldsymbol{b}_{\boldsymbol{k}}\left(\boldsymbol{t}\right) + \varepsilon\left(\boldsymbol{u}_{\boldsymbol{k}}\right) = \boldsymbol{W}_{\boldsymbol{k}}^{\boldsymbol{T}}\cdot\boldsymbol{\sigma}\left(\boldsymbol{u}_{\boldsymbol{k}}\right) + \varepsilon\left(\boldsymbol{u}_{\boldsymbol{k}}\right) + \widehat{\boldsymbol{W}}_{\boldsymbol{k}}^{\boldsymbol{T}}\cdot\boldsymbol{\sigma}^{'}\left(\boldsymbol{u}_{\boldsymbol{k}}\right)\widetilde{\boldsymbol{W}}_{\boldsymbol{ik}}^{\boldsymbol{T}}\boldsymbol{\sigma}_{\boldsymbol{i}}\left(\boldsymbol{w}_{\boldsymbol{k}}\right) - \widetilde{\boldsymbol{W}}_{\boldsymbol{k}}^{\boldsymbol{T}}\cdot\boldsymbol{\sigma}^{'}\left(\boldsymbol{u}_{\boldsymbol{k}}\right)\widehat{\boldsymbol{W}}_{\boldsymbol{ik}}^{\boldsymbol{T}}\boldsymbol{\sigma}_{\boldsymbol{i}}\left(\boldsymbol{w}_{\boldsymbol{k}}\right) - \boldsymbol{w}_{\boldsymbol{k}} \quad (25)$$

By combining Eqs. (19) and (21), the above equation can be written as:

$$\boldsymbol{d}_{\boldsymbol{k}}\left(t\right) = \boldsymbol{\tau}_{\boldsymbol{k}} + \widehat{\boldsymbol{W}}_{\boldsymbol{k}}^{\boldsymbol{T}}\cdot\boldsymbol{\sigma}^{'}\left(\boldsymbol{u}_{\boldsymbol{k}}\right)\widetilde{\boldsymbol{W}}_{\boldsymbol{ik}}^{\boldsymbol{T}}\boldsymbol{\sigma}_{\boldsymbol{i}}\left(\boldsymbol{w}_{\boldsymbol{k}}\right) - \widetilde{\boldsymbol{W}}_{\boldsymbol{k}}^{\boldsymbol{T}}\cdot\boldsymbol{\sigma}^{'}\left(\boldsymbol{u}_{\boldsymbol{k}}\right)\widehat{\boldsymbol{W}}_{\boldsymbol{ik}}^{\boldsymbol{T}}\boldsymbol{\sigma}_{\boldsymbol{i}}\left(\boldsymbol{w}_{\boldsymbol{k}}\right) - \boldsymbol{w}_{\boldsymbol{k}}$$

Namely, we have:

$$\boldsymbol{\tau}_{\boldsymbol{k}} = \boldsymbol{w}_{\boldsymbol{k}} - \widehat{\boldsymbol{W}}_{\boldsymbol{k}}^{\boldsymbol{T}}\cdot\boldsymbol{\sigma}^{'}\left(\boldsymbol{u}_{\boldsymbol{k}}\right)\widetilde{\boldsymbol{W}}_{\boldsymbol{ik}}^{\boldsymbol{T}}\boldsymbol{\sigma}_{\boldsymbol{i}}\left(\boldsymbol{w}_{\boldsymbol{k}}\right) + \widetilde{\boldsymbol{W}}_{\boldsymbol{k}}^{\boldsymbol{T}}\cdot\boldsymbol{\sigma}^{'}\left(\boldsymbol{u}_{\boldsymbol{k}}\right)\widehat{\boldsymbol{W}}_{\boldsymbol{ik}}^{\boldsymbol{T}}\boldsymbol{\sigma}_{\boldsymbol{i}}\left(\boldsymbol{w}_{\boldsymbol{k}}\right) + \boldsymbol{d}_{\boldsymbol{k}}\left(t\right) \quad (26)$$

Then, it is proved.

Lemma2: The norm of model non-match term ***d(t)*** is bounded, and its upper bound is,

$$\|\boldsymbol{d}\left(t\right)\| \le a_1\|\widetilde{\boldsymbol{W}}\|_{\boldsymbol{F}} + a_2\|\widetilde{\boldsymbol{W}}_{\boldsymbol{i}}\|_{\boldsymbol{F}}^{2} + a_3\|\widetilde{\boldsymbol{W}}_{\boldsymbol{i}}\|_{\boldsymbol{F}} + a_4 \quad (27)$$

where a_1, a_2, a_3, a_5 is a constant that can be computed.

This can be proved as follow:

From the definition of $\boldsymbol{d}\left(t\right)$ in Eq. (19), we easily know that

$$\|\boldsymbol{d}\left(t\right)\| \le \|\widetilde{\boldsymbol{W}}\|_{\boldsymbol{F}}\|\boldsymbol{\sigma}^{'}\left(\cdot\right)\|\|\boldsymbol{W}_{\boldsymbol{i}}\|_{\boldsymbol{F}}\|\boldsymbol{\sigma}_{\boldsymbol{i}}\left(\cdot\right)\| + \|\boldsymbol{b}\left(t\right)\| + \|\boldsymbol{\varepsilon}\left(\boldsymbol{u}\right)\| \le \|\widetilde{\boldsymbol{W}}\|_{\boldsymbol{F}}\boldsymbol{a}_1 + \|\boldsymbol{b}\left(t\right)\| + \boldsymbol{\epsilon}_{\boldsymbol{N}} \quad (28)$$

where $\boldsymbol{a}_1 = \|\dot{\boldsymbol{\sigma}}\left(\cdot\right)\|\boldsymbol{W}_{iM}\|\boldsymbol{\sigma}_{\boldsymbol{i}}\left(\cdot\right)\|$

From the definition of $\boldsymbol{b}\left(t\right)$ in Eq. (20), and Eq. (23), we know that,

$$\begin{aligned}\|\boldsymbol{b}\left(t\right)\| \le{}& \boldsymbol{W}_{\boldsymbol{M}}\|\dot{\boldsymbol{\sigma}}\left(\cdot\right)\|\varepsilon_{\boldsymbol{Ni}} + \boldsymbol{W}_{\boldsymbol{M}}\frac{1}{2}\|\ddot{\boldsymbol{\sigma}}\left(\cdot\right)\|\|\widetilde{\boldsymbol{W}}\|_{\boldsymbol{F}}^{2}\|\boldsymbol{\sigma}_{\boldsymbol{i}}\left(\cdot\right)\|^{2}\\ &+ \boldsymbol{W}_{\boldsymbol{M}}\|\ddot{\boldsymbol{\sigma}}\left(\cdot\right)\|\|\widetilde{\boldsymbol{W}}_{\boldsymbol{i}}\|\boldsymbol{\sigma}_{\boldsymbol{i}}\left(\cdot\right)\|\varepsilon_{\boldsymbol{Ni}} + \boldsymbol{W}_{\boldsymbol{M}}\frac{1}{2}\|\ddot{\boldsymbol{\sigma}}\left(\cdot\right)\|\varepsilon_{\boldsymbol{Ni}}\\ &+ \varepsilon_{\boldsymbol{N}} = \boldsymbol{a}_2\|\widetilde{\boldsymbol{W}}\|_{\boldsymbol{F}}^{2} + \boldsymbol{a}_3\|\widetilde{\boldsymbol{W}}_{\boldsymbol{i}}\|_{\boldsymbol{F}} + \boldsymbol{a}_4\end{aligned} \quad (29)$$

where $\boldsymbol{a}_2 = \boldsymbol{W}_{\boldsymbol{M}}\frac{1}{2}\|\ddot{\boldsymbol{\sigma}}\left(\cdot\right)\|\|\boldsymbol{\sigma}_{\boldsymbol{i}}\left(\cdot\right)\|^{2}$, $\boldsymbol{a}_3 = \boldsymbol{W}_{\boldsymbol{M}}\|\ddot{\boldsymbol{\sigma}}\left(\cdot\right)\|\|\boldsymbol{\sigma}_{\boldsymbol{i}}\left(\cdot\right)\|\varepsilon_{\boldsymbol{Ni}}$,

$$\boldsymbol{a}_4 = \boldsymbol{W}_{\boldsymbol{M}}\|\dot{\boldsymbol{\sigma}}\left(\cdot\right)\|\varepsilon_{\boldsymbol{Ni}} + \boldsymbol{W}_{\boldsymbol{M}}\frac{1}{2}\|\ddot{\boldsymbol{\sigma}}\left(\cdot\right)\|\varepsilon_{\boldsymbol{Ni}} + \varepsilon_{\boldsymbol{N}}$$

From Eq. (28) and (29), Lemma2 was approved.

The adaptive law of neural network compensator is designed as follows:

$$\begin{cases}\dot{\widehat{\boldsymbol{W}}}_{\boldsymbol{k}} = -\boldsymbol{S}\boldsymbol{\sigma}^{'}\left(\boldsymbol{u}_{\boldsymbol{k}}\right)\boldsymbol{W}_{\boldsymbol{ik}}{}^{\boldsymbol{T}}\boldsymbol{\sigma}_{\boldsymbol{i}}\left(\boldsymbol{w}_{\boldsymbol{k}}\right)\boldsymbol{r}_{\boldsymbol{k}} - \boldsymbol{K}_1 S\|\boldsymbol{r}\|\widehat{\boldsymbol{W}}_{\boldsymbol{k}}\\ \dot{\widehat{\boldsymbol{W}}}_{\boldsymbol{ik}} = T\boldsymbol{\sigma}_{\boldsymbol{i}}\left(\boldsymbol{w}_{\boldsymbol{k}}\right)\boldsymbol{r}_{\boldsymbol{k}}\boldsymbol{W}_{\boldsymbol{k}}{}^{\boldsymbol{T}}\boldsymbol{\sigma}^{'}\left(\boldsymbol{u}_{\boldsymbol{k}}\right) - K_1 T\|\boldsymbol{r}\|\widehat{\boldsymbol{W}}_{\boldsymbol{ik}} - \boldsymbol{K}_2 T\|\boldsymbol{r}\|\|\widehat{\boldsymbol{W}}_{\boldsymbol{i}}\|_{\boldsymbol{F}}\widehat{\boldsymbol{W}}_{\boldsymbol{ik}}\end{cases} \quad (30)$$

where, $S = S^{\mathrm{T}}$,$T = T^{\mathrm{T}}$ is a positive definite constant symmetric matrix, $K_1, K_2 > 0$. $\widehat{W}_k$ and $\widehat{W}_{ik}$ is the k^{th} group weight vector in the *n* group weight vector of the compensator ($1 \le k \le n$).

System stability analysis

Given that the ideal trajectory is $\boldsymbol{q}_{\boldsymbol{d}}(\boldsymbol{t}) \in \boldsymbol{R}^{\boldsymbol{n}}$, the tracking error is:

$$\boldsymbol{e}\left(\boldsymbol{t}\right) = \boldsymbol{q}_{\boldsymbol{d}}\left(\boldsymbol{t}\right) - \boldsymbol{q}(\boldsymbol{t}) \quad (31)$$

Define the error function as:

$$\boldsymbol{r} = \dot{\boldsymbol{e}} - \boldsymbol{\Lambda}\boldsymbol{e} \quad (32)$$

where $\boldsymbol{\Lambda} = \boldsymbol{\Lambda}^{\boldsymbol{T}} > 0$.

Define $\dot{\boldsymbol{q}}_{\boldsymbol{r}} = \boldsymbol{r} + \dot{\boldsymbol{q}}$, then $\ddot{\boldsymbol{q}}_{\boldsymbol{r}} = \boldsymbol{r} + \ddot{\boldsymbol{q}}$, $\dot{\boldsymbol{q}}_{\boldsymbol{r}} = \dot{\boldsymbol{q}}_{\boldsymbol{d}} + \boldsymbol{\Lambda}\boldsymbol{e}$, $\ddot{\boldsymbol{q}}_{\boldsymbol{r}} = \ddot{\boldsymbol{q}}_{\boldsymbol{d}} + \boldsymbol{\Lambda}\dot{\boldsymbol{e}}$

According to the system dynamic Eqs. (1), (6) and (7), there is

$$\boldsymbol{M}\dot{\boldsymbol{r}} = \boldsymbol{M}\left(\dot{\boldsymbol{e}} - \boldsymbol{\Lambda e}\right) = -\boldsymbol{V_m r} - \boldsymbol{\tau} + \boldsymbol{f} \tag{33}$$

where $\boldsymbol{f} = \boldsymbol{M}\ddot{\boldsymbol{q}}_{\boldsymbol{r}} + \boldsymbol{V}_{\boldsymbol{m}}\dot{\boldsymbol{q}}_{\boldsymbol{r}} + \boldsymbol{G}, \boldsymbol{f}$ represents model information.

Take the ideal control signal as:

$$\boldsymbol{w} = \widehat{\boldsymbol{f}} + \boldsymbol{K}_{\boldsymbol{r}}\boldsymbol{r} - \boldsymbol{v} \tag{34}$$

where $\widehat{f}$ refers to the estimation of f, and the error is $\widetilde{f} = \widehat{f} - f$, $\|\widehat{f}\| \leq f_M(x)$, v is a robust term used to overcome modeling uncertainties.

The robust control law is:

$$\boldsymbol{v}\left(\boldsymbol{t}\right) = -\boldsymbol{f}_{\boldsymbol{M}}(\boldsymbol{x})\mathrm{sgn}(\boldsymbol{r}) \tag{35}$$

where $\mathrm{sgn}(\cdot)$ is a sign function.

By substituting the control law (18) and Eq. (34) into Eq. (33), we have:

$$\boldsymbol{M}\dot{\boldsymbol{r}} = -\boldsymbol{V_m r} - \boldsymbol{K_v r} + \left[\left\{\widehat{\boldsymbol{W}}\right\}^T \cdot \left\{\boldsymbol{\sigma}'\left(\boldsymbol{u}\right)\right\}\right] \cdot \left[\left\{\widehat{\boldsymbol{W}}_{\boldsymbol{i}}\right\}^T \cdot \left\{\boldsymbol{\sigma}_{\boldsymbol{i}}\left(\boldsymbol{w}\right)\right\}\right] - \left[\left\{\widetilde{\boldsymbol{W}}\right\}^T \cdot \left\{\boldsymbol{\sigma}'\left(\boldsymbol{u}\right)\right\}\right] \cdot \left[\left\{\widehat{\boldsymbol{W}}_{\boldsymbol{i}}\right\}^T \cdot \left\{\boldsymbol{\sigma}_{\boldsymbol{i}}\left(\boldsymbol{w}\right)\right\}\right] - \boldsymbol{d}\left(\boldsymbol{t}\right) + \boldsymbol{f} - \widehat{\boldsymbol{f}} + \boldsymbol{v} \tag{36}$$

Given the robotic system (1), it is acceptable to ignore the friction terms and disturbance terms[36]. Assuming that the neural network has a bounded ideal weight, $\|\mathrm{W}\|_{\mathrm{F}} \leq \mathrm{W_M}, \|\mathrm{W_i}\|_{\mathrm{F}} \leq \mathrm{W_{iM}}$, select the controller Eq. (34) and the neural network compensator Eq. (17). Taking the adaptive law of neural network as Eq. (30), the closed-loop tracking control error of the control system is globly uniformly ultimately bounded stable, and the tracking error decreases with the increase of the gain Kv.

Simulation and experiment

Simulation model

The mathematical model of the double-joint tea picking mechanical arm is as Eqs. (37)–(40):

$$M(q)\ddot{q} + V_m\left(q, \dot{q}\right)\dot{q} + G\left(q\right) = \tau \tag{37}$$

$$D\left(q\right) = \begin{bmatrix} p_1 + p_2 + 2p_3 cosq_2 & p_2 + p_3 cosq_2 \\ p_2 + p_3 cosq_2 & p_2 \end{bmatrix} \tag{38}$$

$$C\left(q, \dot{q}\right) = \begin{bmatrix} -p_3\dot{q}_2 sinq_2 & -p_3(\dot{q}_1 + \dot{q}_2) sinq_2 \\ -p_3\dot{q}_1 sinq_2 & 0 \end{bmatrix} \tag{39}$$

$$G\left(q\right) = \begin{bmatrix} p_4 cosq_1 + p_5 cos(q_1 + q_2) \\ p_5 cos(q_1 + q_2) \end{bmatrix} \tag{40}$$

The actual parameters used in the simulation are:

$$\boldsymbol{p} = [p_1, p_2, p_3, p_4, p_5] = [3.5, 0.66, 0.93, 3.04, 0.9]$$

It is reasonable to assume that the initial states of the robot are all at the zero position, and the parts outside the dead zone are linear functions[37]. The dead zone parameter is $d_+ = 10, d_- = -10, h\left(\boldsymbol{u}\right) = \boldsymbol{u} - d_+, g\left(\boldsymbol{u}\right) = \boldsymbol{u} + d_-$. The expected trajectory is: $q_d\left(t\right) = \begin{bmatrix} 1.5 sin(0.3\pi t) \\ cos(0.3\pi t) \end{bmatrix}$. The parameters of the neural network controller are set to $\Lambda = \Lambda^T = \mathrm{diag}\ [5.0], \boldsymbol{K_V} = 20, \boldsymbol{K_r} = 0, S = \begin{bmatrix} 400 & 0 \\ 0 & 400 \end{bmatrix}, T = \begin{bmatrix} 400 & 0 \\ 0 & 400 \end{bmatrix}$, The central value c of the m-RBF neural network is the value uniformly distributed within the input value range of the RBF network. $b = 1$ $c_1 = c_2 = \begin{bmatrix} -50 & -28 & -15 & 0 & 15 & 28 & 50 \\ -50 & -28 & -15 & 0 & 15 & 28 & 50 \end{bmatrix}$, The initial value of each element in the weight of m-RBF neural network is 20. The simulink simulation model of the system is shown in Fig. 4. The simulation is conducted on an experimental computer, the parameters of which is as follow: CPU Intel(R) Core(TM) i7-4790, @3.60 GHz, number of cores: 8, memory: 4 GB.

Simulation result

Tracking performance

Simulation results are shown in Figs. 5, 6 and 7. It is illustrated in Fig. 5 that the tracking performance of the control model is rather exact. The red line is the expected trajectory, while the blue ones denote the tracking trajectory. Both in the position tracking of link 1 and link 2, our control model could track the target within 1 second and copy the expected trajectory with nearly no bias once the expected trajectory is tracked.

Control input analysis

The control input of the control system is shown in Fig. 6. The ideal control input without consideration of the dead zone is shown in Fig. 6a, in which situation it is assumed that the links are deal ones that has no dead zone. Therefore its control input is with rather less fluctuation, especially for link 2.

However, the impact of the dead zone shouldn't be neglected, because it has a sever side effect on the tracking performance of the manipulator, such as impairing tracking speed and accuracy. In this work, the dead zone is

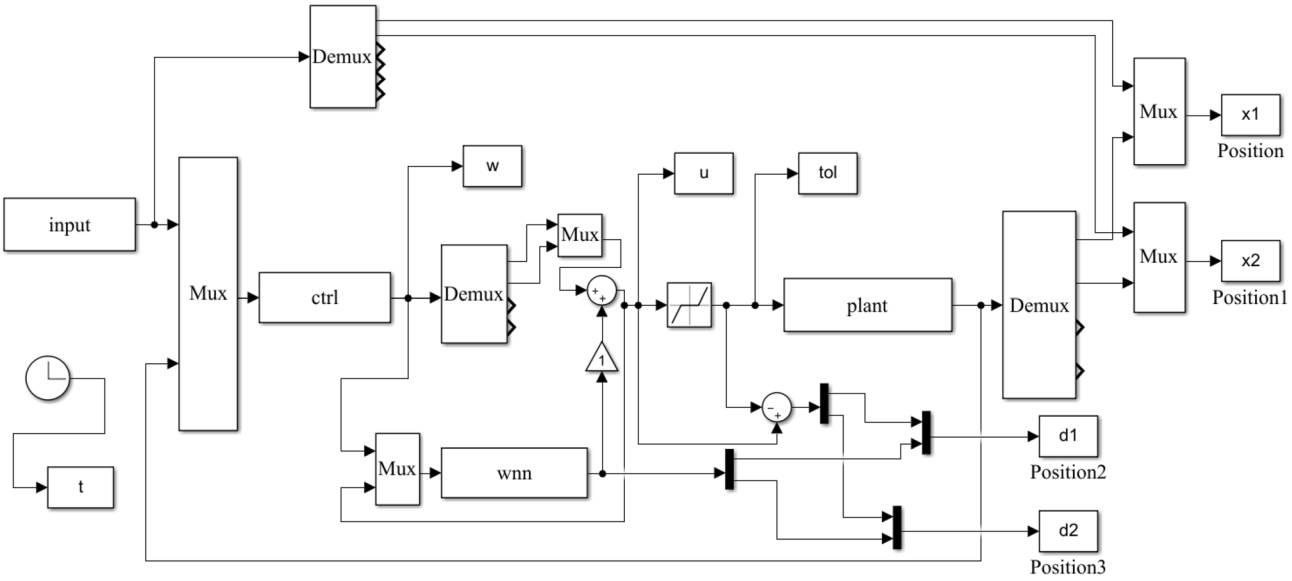

Fig. 4. Simulink simulation model of the system.

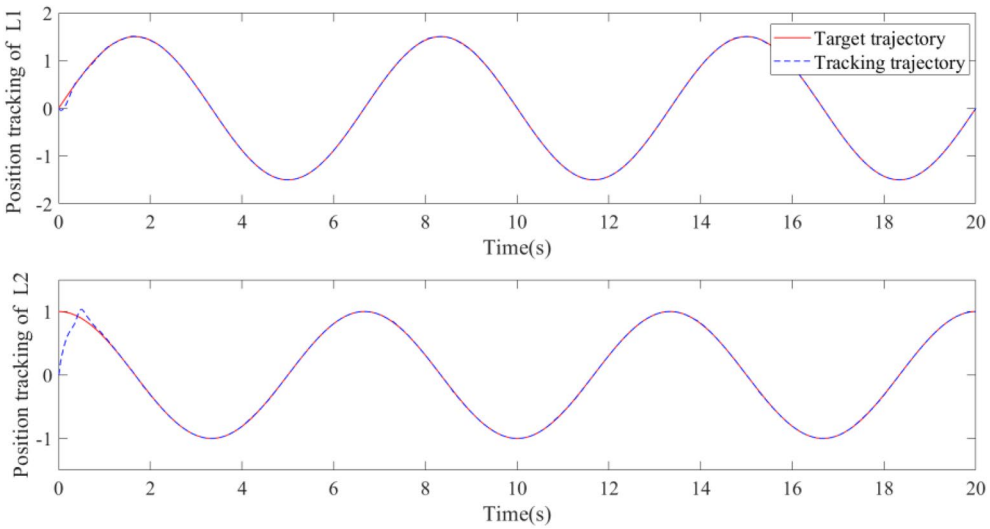

Fig. 5. Position tracking. L1 denotes Link 1, L2 denotes Link2.

well modeled with m-RBF, the compensation for the dead zone is integrated into the control input. As shown in Fig. 6b, both control input for joint 1 and joint 2 become more fluctuated due to the introduction of the dead zone model. Even so, the variation range is still be confined between ±50 for joint 1 and ±30 for joint 2. Compared with ideal control input variation range ([− 20, 40] for link 1, [− 10,5] for link 2), The variation range increments is still tolerable and physically achievable.

Dead zone estimation

As shown in Fig. 1, we know that the characters of dead zone nonlinearity can be estimated by equation (2). When the manipulator is running at a high speed, the nonlinearity of dead zone could be shown with a trapezoidal wave shown in Fig. 7 with red line. From the figure, it can be seen that our m-RBF model coule model the characotoristics of the nonlinearity of dead zone perfectly. Because the m-RBF is training in an online manner, the Net needs 1~2 seconds to converse to stable state when running for the first time. Although, there is small overshoot for link 1 when it steps in each dead zone, but the model could convers quickly to the target. As a whole, the proposed model could estimate the dead zone well.

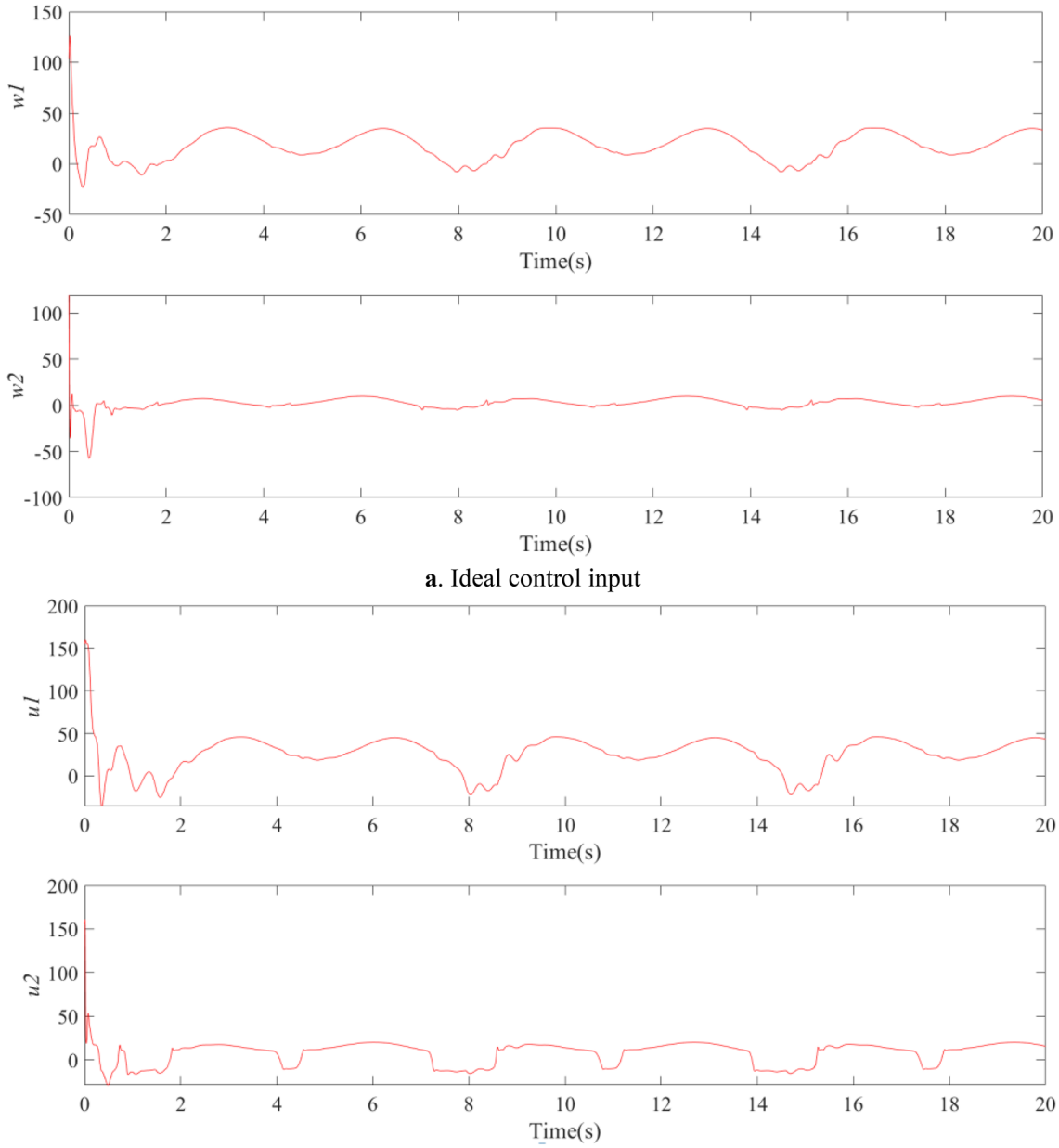

Fig. 6. Control input.

Severe situation tracking performance

In order to simulate the actual picking process as much as possible, we set two more jumping target trajectories: a sine wave of higher frequency (see in Fig. 8), and a compound curve of sine wave and square wave as Eq. (41), see in Fig. 9.

$$f(x) = 2sin(0.2\pi x) + 8square(0.2\pi x) \tag{41}$$

where *square*(·) is a MATLab built-in function that produces a square wave.

As shown in Fig. 8, the target trajectory of higher frequency sine wave simulates the situation of high-speed continuous turning motion. And our mehtod could tracking this kind of motion precisely within about 1s, except for a slight error at the turning point. As shown in Fig. 9, the target trajectory composed of a sine wave and a square wave simulates the situations of sudden start, emergency stop and reverse return. As is well known, it is relatively difficult to track step signals. The simulation result indicates that the proposed method could handle the tracking of these motions well. This "jumping" motion could also be tracked within 1s. The simulation of severe situation indicates that the proposed control method is suitable for the robot application of tea picking.

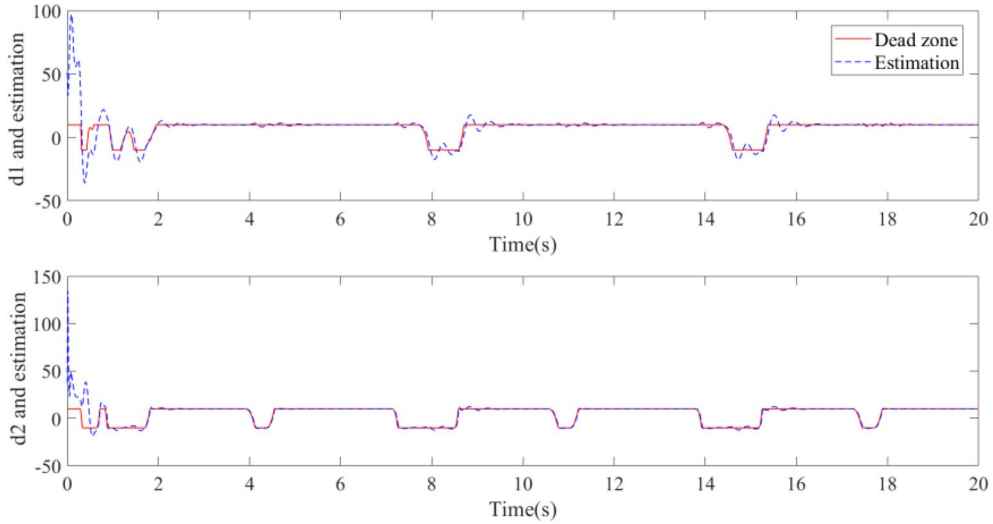

Fig. 7. Dead zone and its estimation.

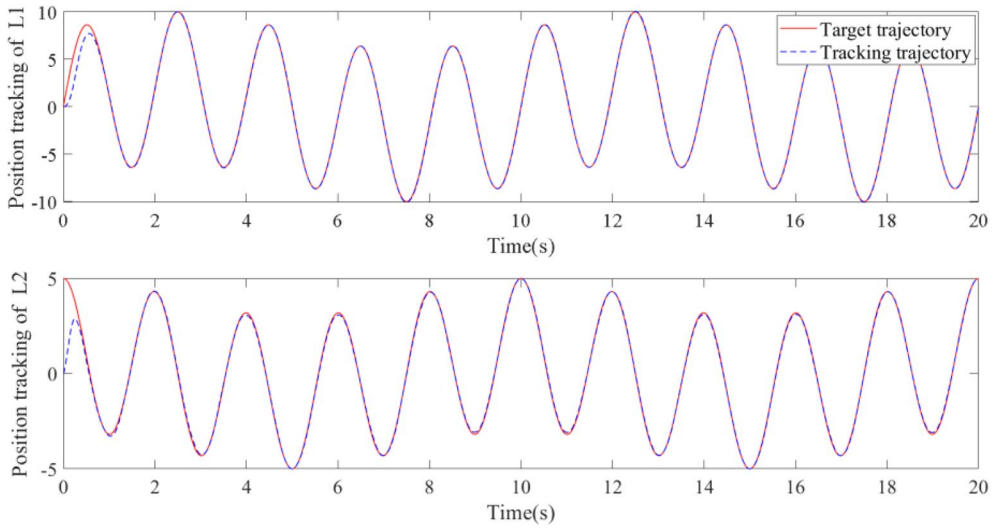

Fig. 8. Tracking of a sine wave of higher frequency.

Experiment

In order to verify the effectiveness of the proposed algorithm, an experiment on tea picking is conducted using the robot test platform. The platform mainly consists of Vision system, Manipulator, Picking mechanism, Computer control center, see in Fig. 10. Some tea tree branches are picked and kept in a bucket with some water in it, which analogies the tea tree canopy. The vision system detects the tender tea shoots and provides their three-dimensional coordinate information to robot control system as the target location. The optimal running path is planned and calculated according to the target points by control system. Then the manipulator will be controlled to pick tea shoot along the running path.

In our previous experiments, generally, the arm would be chattering or get stuck when it encountered a dead zone. When the proposed algorithm is applied to the robot test platform, the running of the arm becomes smoother. To illustrate our algorithm's effectiveness, two sets of comparative experiments are conducted: one group used the traditional control method, and the other group added the death compensation control method proposed in this paper. Each group includes 100 times picking test, tracking one picking point in each test. The breakdown get chattering or stuck) are recorded and a score is given for each control mode, as shown in Table 1.

The initial score for each control mode is 100, and a penalty is given for each failure. The penalty is determined by the kind of the breakdown, − 0.3 for chattering and − 0.7 for being stuck. In one test, multiple chattering may occur, but only once is recorded to avoid a negative general score.

From Table 1, we know that it is very prone to be breakdown under the traditional control mode, while our control method based on dead zone estimation using RBF could improve this defect significantly, with a

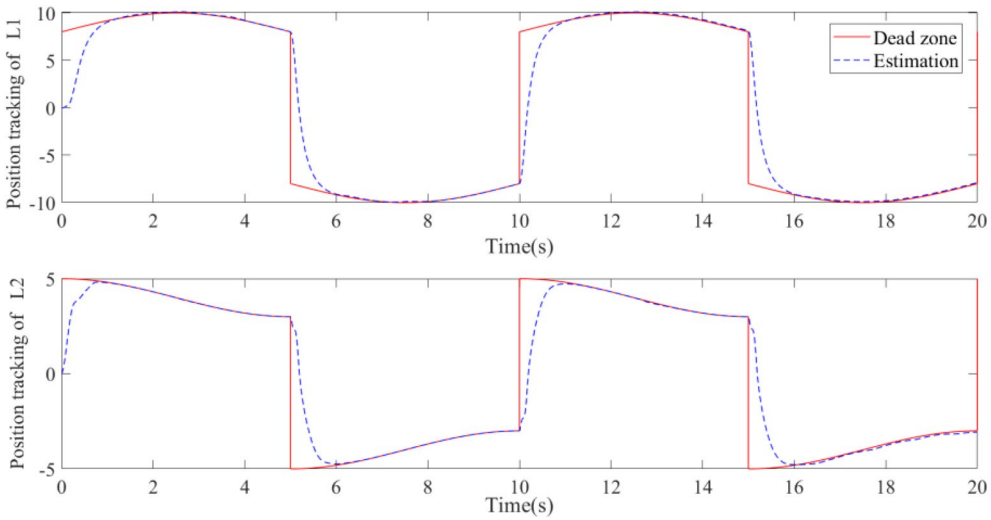

Fig. 9. Tracking a compound curve of sine wave and square wave.

Fig. 10. Robot platform and experiment.

Control mode	Breakdown times		P	S
	Ch	St		
Traditional method	51	27	−42.3	57.7
The proposed method	9	2	−4.7	95.3

Table 1. Comparative experiment result. Ch and St denote the breakdown of chattering and being stuck respectively; P is penalty, S is general score of different control mode.

high score of 95.3. That is to say, the proposed method for the estimation of manipulator's dead zone and the compensation of its control is effective.

Conclusion and discussion

As one of the main nonlinear characteristics, dead zone nonlinearity has an important effect on the performance of power transmission system. In this study, m-RBF neural network adaptive compensator and adaptive control law were designed for the nonlinear link and dead zone characteristics of feed-forward channel of tea picking robot system. Through the stability analysis of the system, it can be seen that by selecting the controller, neural network compensator and neural network adaptive law designed in this paper, the closed-loop tracking error of the control system is always stable and bounded eventually, and the tracking error decreases with the increase of the gain. Compared with other works, Zhou[30], He[25], and Bensidhoum[32], our method has globally uniformly ultimately bounded stability, which outperforms their semi-globally uniformly ultimately bounded, semi-globally uniformly bounded and asymptotically stable respectively.

To visually demonstrate the performance of the control system, taking the two joints manipulator as an example, the system simulation model is established in MATLAB/Simulink. The simulation results show that the m-RBF neural network has a good approximation performance for the nonlinear system, and can approximate the dead zone nonlinear characteristics well. The control system based on the neural network model can realize the stable tracking control of a manipulator better, and has a strong ability to restrain the load disturbance.

Finally, the proposed algorithm was applied to a six-axis manipulator to carry out tea picking experiments. Compared with the traditional PID control methods (Proportional, Integral, Differential control), the proposed method got a higher score of 95.3 in the comprehensive evaluation, and could effectively deal with the problems caused by the dead zone of the manipulator, making the operation of the manipulator more smooth and efficient.

Because of the local approximation property of m-RBF neural network, the learning speed is fast and the local minimum problem can be avoided, which is especially suitable for the application requirements of real-time control. Therefore, the neural network control scheme of the tea picking manipulator composed of m-RBF network has good control precision, robustness and adaptability. As shown in section "Simulation result", it still takes about 1 s to adjust before tracking stably, which is not good for real-time tracking applications. Therefore, it is necessary to further study the dead zone compensation algorithm and network structure in the future to improve the compensation and tracking performance.

Data availability

Data is provided within the manuscript or supplementary information files, further inquiries can be directed to the corresponding author.

Received: 14 January 2025; Accepted: 7 July 2025
Published online: 21 August 2025

References

1. Fu, L. & Xu, Q. Research on the status quo and promotion strategy of Chinese tea industry. *Chin. Business Theory* **06**(2), 221–223 (2019).
2. Han, Y. et al. Research on the development status of tea picking machinery at home and abroad. *Chin. J. Agric. Mech.* **35**(02), 20–24 (2014).
3. Vu, H., Le, T., Tran, T. & Nguyen, T. A vision-based method for automatizing tea shoots detection. In: IEEE International Conference on Image Processing, Melbourne, VIC, Australia, 3775–3779 (2013).
4. Thangavel, S. & Mursi, M. A semi-automated system for smart harvesting of tea leaves. In: 4th International Conference on Advanced Computing and Communication Systems (ICACCS), Coimbatore, India, 1–10 (2017).
5. Karunasena, G. & Priyankara, H. Tea bud leaf identification by using machine learning and image processing. *Techniques* **11**(8), 624–628 (2020).
6. Jain, S. K., et al. IoT based smart tea leaves plucker with two revolute type planar manipulator. In: IEEE International Students' Conference on Electrical, Electronics and Computer Science (SCEECS), Bhopal, India, 1–6 (2018).
7. Motokura, K., Takahashi, M., Ewerton, M. & Peters, J. Plucking motions for tea harvesting robots using probabilistic movement primitives. *IEEE Robot. Autom. Lett.* **5**(2), 3275–3282 (2020).
8. Yang, F. Z., Yang, L. L., Tian, Y. N. & Yang, Q. Recognition of the tea sprout based on color and shape features. *Trans. Chin. Soc. Agric. Mach.* **40**, 119–123 (2009).
9. Jin, X. J., Chen, Y., Zhang, H., Sun, Y. X. & Chen, J. High-quality tea flushes detection under natural conditions using computer vision. *JDCTA Int. J. Digit. Content Technol. Appl.* **6**(8), 600–606 (2012).
10. Wang, J., Zeng, X. Y. & Liu, J. B. Three-dimensional modeling of tea-shoots using images and models. *Sensors* **11**, 3803–3815 (2011).
11. Qi, F., Xie, Z. & Tang, Z. Related study based on otsu watershed algorithm and new squeeze-and-excitation networks for segmentation and level classification of tea buds. *Neural Process. Lett.* **53**(3), 2261–2275 (2021).
12. Lv, J., Xia, H. K., Fang, M. R. & Zhou, L. Z. Research on intelligent identification of tea sprouts state based on alex net. *J. Heilongjiang Bayi Agric. Univ.* **31**(2), 72–78 (2019).
13. Yang, H. L. et al. Tender tea shoots recognition and positioning for picking robot using improved yolo-v3 model. *IEEE Access.* **7**, 180998–181011 (2019).
14. Yang, H. L. et al. Computer vision-based high-quality tea automatic plucking robot using Delta parallel manipulator. *Comput. Electron. Agric.* **181**, 105946–105955 (2021).
15. Li, Y. L. et al. In-field tea shoot detection and 3D localization using an RGB-D camera. *Comput. Electron. Agric.* **185**, 106149 (2021).
16. Gong, T. & Wang, Z. L. A tea tip detection method suitable for tea pickers based on YOLOv4 network. In *2021 3rd International Symposium on Robotics & Intelligent Manufacturing Technology (ISRIMT)* (eds Gong, T. & Wang, Z. L.) 264–268 (IEEE, 2021).
17. Chen, Y. T. & Chen, S. F. Localizing plucking points of tea leaves using deep convolutional neural networks. *Comput. Electron. Agric.* **171**, 105298 (2020).
18. Chen, M. T. *Recognition and Localization of Famous Tea Buds Based on Computer Vision* (Qingdao University of Science and Technology, 2019).
19. Wei, J. J. *Research on Bud Identification Method for Mechanized Picking of Famous Tea* (Nanjing Forestry University, 2012).

20. Sun, X. F. *Research on the Method of Identifying New Shoot and Determining Center Point for Intelligent Picking of Famous Tea* (Nanjing Forestry University, 2014).
21. Zhao, Z. J. et al. Adaptive quantized control of flexible manipulators subject to unknown dead zones. *IEEE Trans. Syst. Man Cybern. Syst.* **53**(10), 6438–6447 (2023).
22. Charandabi, B. A., Salmasi, F. R. & Sedigh, A. K. Improved dead zone modification for robust adaptive control of uncertain linear systems described by input-output models with actuator faults. *IEEE Trans. Autom. Control* **56**(4), 863–867 (2011).
23. Liu, Z., Lai, G. Y., Zhang, Y. & Philip Chen, C. L. Adaptive fuzzy tracking control of nonlinear time-delay systems with dead-zone output mechanism based on a novel smooth model. *IEEE Trans. Fuzzy Syst.* **23**(6)**,** 1998–2011(2015).
24. Zhao, Z. J., Ahn, C. K. & Li, H. X. Dead zone compensation and adaptive vibration control of uncertain spatial flexible riser systems. *IEEE/ASME Trans. Mechatron.* **25**(3), 1398–1408 (2020).
25. He, W., Huang, B., Dong, Y., Li, Z. & Su, C. Y. Adaptive neural network control for robotic manipulators with unknown deadzone. *IEEE Trans. Cybern.* **48**(9), 2670–2682 (2018).
26. Wang, X. S., Su, C. Y. & Hong, H. Robust adaptive control of a class of nonlinear systems with unknown dead-zone. *Automatica* **40**(3), 407–413 (2004).
27. Tong, S., Zhang, L. & Li, Y. Observed-based adaptive fuzzy decentralized tracking control for switched uncertain nonlinear large-scale systems with dead zones. *IEEE Trans. Syst. Man Cybern. Syst.* **46**(1), 37–47 (2016).
28. Pei, L., et al. A robust deadzone compensation method against parameter variations based on kalman filter and neural networks. *IECON 2021 – 47th Annual Conference of the IEEE Industrial Electronics Society*, Toronto, Canada, 1–6 (2021).
29. Lewis, F. L., Liu, K. & Yesidirek, A. Neural net robot controller with guaranteed tracking performance. *IEEE Trans. Neural Netw.* **6**(3), 703–715 (1995).
30. Zhou, Q., Zhao, S. Y. & Li, H. Y. Adaptive neural network tracking control for robotic manipulators with dead zone. *IEEE Trans. Neural Netw. Learn. Syst.* **30**(12), 3611–3620 (2019).
31. Lu, Y., Liu, J. K. & Sun, F. C. Actuator nonlinearities compensation using rbf neural networks in robot control system. In: Proc. 4th CESA Multi-conference on Computational Engineering in Systems Application, Beijing, China. **1,** 231–238 (2006).
32. Bensidhoum, T. & Bouakrif, F. Adaptive P-type iterative learning radial basis function control for robot manipulators with unknown varying disturbances and unknown input dead zone. *Int. J. Robust Nonlinear Control* **30**, 4075–4094 (2020).
33. Ge, S. S. Robust adaptive NN feedback linearization control of nonlinear systems. *Int. J. Syst. Sci.* **27**(2), 1327–1338 (1996).
34. Chiu, S. F. Fuzzy model identification based on cluster estimation. *J. Intell. Fuzzy Syst.* **2**(3), 749–755 (1994).
35. Wang, Y. N. *Robot Intelligent Control Engineering* 226–229 (Science Press, 2003).
36. Zhao, H., Liu, W. Y., Chen, X. L. & Sun, H. Adaptive robust constraint-following control for under actuated unmanned bicycle robot with uncertainties. *ISA Trans.* **143**(04), 144–155 (2023).
37. Peng, H. J., Li, N., Jiang, D. H. & Li, F. Soft robot fast simulation via reduced order extended position based dynamics. *Robot. Auton. Syst.* **175**(05), 204–215 (2024).

Acknowledgements

This work was supported mainly by the Key Laboratory of Modern Agricultural Equipment, Ministry of Agriculture and Rural Affairs, P.R., China; the open subject from Key Laboratory of Agricultural Equipment Technology for Hilly and Mountainious Areas under Grant 20224LOP04; the Modern Agricultural Technology System of Tea Industry (Grant No. CARS-19, China), The Innovating Program of the Chinese Academy of Agricultural Science, China; The Key Laboratory for Crop Production and Smart Agriculture of Yunnan Province (2023ZHNY06).

Author contributions

Conceptualization, Y.H.; Writing—original draft， Y.H.; methodology, Y.H.; software, Y.S.; validation, Y.S.; formal analysis, Y.H.; project administration, Y.Y.; funding acquisition, Y.Y.; Experiment, Y. H., Y.S., Y.Y., C.Z. All authors have read and agreed to the published version of the manuscript.

Declarations

Competing interests

The authors declare no competing interests.

Additional information

Correspondence and requests for materials should be addressed to Y.H., Z.S. or W.Y.