

Neural-Learning-Based Force Sensorless Admittance Control for Robots with Input Deadzone

Guangzhu Peng, C. L. Philip Chen, *Fellow, IEEE*, Wei He, *Senior Member, IEEE*,
and Chenguang Yang, *Senior Member, IEEE*

Abstract—This paper presents a neural networks based admittance control scheme for robotic manipulators when interacting with the unknown environment in the presence of the actuator deadzone without needing force sensing. A compliant behaviour of robotic manipulators in response to external torques from the unknown environment is achieved by admittance control. Inspired by broad learning system (BLS), a flatted neural network structure using Radial Basis Function (RBF) with incremental learning algorithm is proposed to estimate the external torque, which can avoid retraining process if the system is modelled insufficiently. To deal with uncertainties in the robot system, an adaptive neural controller with dynamic learning framework is developed to ensure the tracking performance. Experiments on the Baxter robot have been implemented to test the effectiveness of the proposed method.

Index Terms—neural networks (NNs); adaptive control; broad learning; force/torque observer; admittance control.

I. INTRODUCTION

IN the past decades, robots have been widely applied in many fields of social life, such as industry, service and education [1]. In these cases, robots are inevitably encountering the unknown environment. Therefore, control of robots in a safe and compliant behaviour during the interactive process has been widely studied. In the literature, hybrid position/force control and impedance control have been the two main approaches for interaction control [2]–[5].

Impedance control aims to impose a desired dynamic behavior on the robot to interact with environments. The desired interaction performance is specified through a impedance model. On the basis of the control causality, admittance control could be regarded as the inverse of impedance control [6]. The admittance control can ensure the robot produce a good interaction performance by adapting the motion to the external force. The admittance control aims to map the

external force measured by the force sensors to the position of the robot end-effector through an admittance model. In general admittance control schemes, the inner-loop is for trajectory tracking, which accepts the trajectory generated by the admittance model in the out-loop as the input signal [7]. In addition, the admittance model varies from different tasks. This characteristic enables the admittance control to be used in many forms to achieve a good physical interaction control performance [8].

In admittance control schemes, force measurement is fundamental and necessary, and the interactive control performance is largely decided by the accuracy of the external force measurement [9]. A common method is to use force sensors to measure external forces. Although the force sensor can provide high precision, it also has shortcomings. First of all, commercial sensors with high precision are often expensive. Secondly, force sensors mounted on robots will change the dynamics of robots [10]. Finally, due to exposure to the complex and dangerous industrial environment, the force sensor will be vulnerable to damage, which is not applicable to certain situations. Therefore, force observer approaches have been widely studied. In [11] [12], the disturbance observer approach was proposed to estimate the external force. In [13], a dynamic state observer was proposed to estimate the contact force. In [14], generalized momentum based observer was used for measuring the contact force with partially known model dynamics. These classical model-based observer methods can not work well without accurate model dynamics.

To solve this problem, many force estimation methods integrating neural networks have been studied [15] [16]. In [15], a NN-improved disturbance observer was proposed with partial dynamics of the robotic link. In [16], NNs were employed to approximate the Coulomb friction and improve the performance of the model-based observer. Although NNs were employed into the above methods, the problem of requiring system dynamics is still unsolved. In [17], an inverse dynamics method combining with multilayer feedforward NNs have been proposed without robot dynamic model. NNs were used to identify the system dynamics and have a good approximation performance. However, this architecture will consumes more time as the number of layers of neural network increases. Compared with traditional multilayer NNs, the random vector functional-link neural network (RVFLNN) is an effective network with good approximation ability and simple structure. Moreover, it can avoid long training time and has fast learning property [18] [19]. Considering this good property, the inverse dynamics observer combined with RVFLNN is proposed for external force. The advantage is that the dynamics of robot system is not required and the training time of the NN structure is shorter.

The control performance is another important factor in

This work was partially supported by Engineering and Physical Sciences Research Council (EPSRC) under Grant EP/S001913, National Natural Science Foundation of China under Grant 61751202, Grant 61751205, Grant U1813203, and Grant U1801262, in part by the Science and Technology Development Fund, Macau SAR (File no. 079/2017/A2, and 0119/2018/A3), in part by the Multiyear Research Grants of University of Macau. *Corresponding author: C. L. Philip Chen.*

G. Peng is with the Department of Computer and Information Science, Faculty of Science and Technology, University of Macau, Macau 999078, China (Email: gz.peng@qq.com).

C. L. Philip Chen is with School of Computer Science and Engineering, South China University of Technology, Guangzhou 510641, China, with the Unmanned System Research Institute, Northwestern Polytechnical University, XiAn, 710072, China, and also with the Faculty of Science and Technology, University of Macau, Macau 999078, China (e-mail: philip.chen@ieee.org).

W. He is with Institute of Artificial Intelligence, University of Science and Technology Beijing, Beijing 100083, China, and also with School of Automation and Electrical Engineering, University of Science and Technology Beijing, Beijing 100083, China (Email:hewei.ac@gmail.com).

C. Yang is with Bristol Robotics Laboratory, University of the West of England, Bristol, BS16 1QY, UK (Email: cyang@ieee.org).

practical control system [20]. In the literature, adaptive control methods incorporating intelligent tools have been widely employed in control systems [21]–[24]. Different from model-based control approaches, these adaptive control methods could deal with the unknown system dynamics by automatically updating laws. Although model-based control usually shows better control performance, it is based on an accurate model [25]. In practice, an accurate model is hardly to be obtained owing to the complexity of mechanical robot system. In addition, the payload on the manipulator could also make the modelling of the robotic arm more difficult [26]. In [27], the RBFNN was employed to deal with uncertainties of both robot system and the object, and a prescribed tracking performance was guaranteed. In [28], NNs were integrated in control design to cope with the uncertain dynamics. In [29], the adaptive fuzzy control method was used to suppress the uncertain dynamics of the system. Although NNs or fuzzy logic can solve the problem of system uncertainties, these intelligent tools inevitably increase the burden of control system. In order to have good approximation performance, the number of neurons and fuzzy rules will increase, which in turn will bring heavy computation burden to the system and affect the control effect. Moreover, when the learning is insufficient, the inaccurate approximation will influence the control performance. However, among the literatures about adaptive NN control, few studies have concerned with the way of solving the problems of the insufficient learning condition and computational burden.

Recently, the broad learning system (BLS) has attracted much attention because of its novel framework without suffering time-consuming training process caused by traditional deep structure, and it has been used in pattern recognition and classification [30] [31]. Considering its novel architecture, the property of BLS could also be employed to NN based control scheme to accumulate the learning experience with node expansion without retraining the model. In [32], it has been pointed out that deterministic learning theory is an efficient way to deal with the unknown dynamics by satisfying the persistent excitation (PE) condition. Based on the above discussion, an adaptive RBFNN learning framework has been developed. Compared with the traditional learning network, the difference is that this adaptive learning framework can not only ensure the approximation ability of NNs under insufficient learning condition, but also will not bring heavy computational burden to the system.

Deadzone is an crucial problem of affecting the dynamic control performance of the system. In practical systems, the deadzone will not only cause the loss of control precision, but also cause system instability. Researchers have proposed many methods to solve this problem. In [33], an adaptive method was incorporated into motion control system for tuning fuzzy logic parameters. In [34], the deadzone nonlinearities in control system were compensated by inverse function. However, due to the complexity and uncertainties of practical actuators, the deadzone functions are difficult to know. Therefore, such model-based compensation methods may not have good performance in practical systems. With the strong approximation ability, NNs have been applied into control system to compensate the deadzone nonlinearity [35]–[38]. Inspired by above works, in this paper, a NN-based force sensorless admittance control scheme for robotic arms with actuator deadzone is proposed. This paper is a continuation of

our previous work [39], and the improvements are given as follows:

(i) Admittance control is adopted to generate a compliant interactive behaviour with considering the dynamics of the unknown environment. A saturation function in joint space is considered to constraint the robot motion.

(ii) A flatted neural network structure using RBF with incremental learning is proposed to estimate the external torque rather than model-based observer.

(iii) A RBFNN-based learning controller with dynamic framework is developed to deal with system uncertainties and compensate the deadzone nonlinearities.

The rest of this paper is organized as follow. Section II expresses the preliminaries. Control design is presented in Section III. Section IV gives the experimental results. Conclusion is drawn in Section V, and appendix is presented in section VI.

II. PRELIMINARIES

A. Problem Formulation

The dynamics of a n–link robotic manipulator is

$$M_q(q)\ddot{q} + C_q(q, \dot{q})\dot{q} + G_q(q) = \tau + J^T f_{ext} \quad (1)$$

where $\ddot{q} \in \mathbb{R}^n$, $\dot{q} \in \mathbb{R}^n$ and $q \in \mathbb{R}^n$ represent joint acceleration, velocity and position, respectively. $M_q(q) \in \mathbb{R}^{n \times n}$ and $C_q(q, \dot{q}) \in \mathbb{R}^{n \times n}$ denote the inertia and coriolis matrices, and $G_q(q) \in \mathbb{R}^n$ is the gravity load. $\tau \in \mathbb{R}^n$ is the joint torque and f_{ext} is the external force. The relationship between the external torque and the force can be written as

$$\tau_{ext} = J^T f_{ext} \quad (2)$$

where $J(q)$ is the manipulator Jacobian matrix.

Property 1: The matrix $M_q(q)$ is symmetric and positive definite [40].

Property 2: The matrix $2C_q(q, \dot{q}) - \dot{M}_q(q)$ is skew-symmetric [40].

The environmental dynamics can be defined as

$$M_E \ddot{x} + C_E \dot{x} + G_E x = -f_{ext} \quad (3)$$

where M_E , C_E and G_E are the coefficient matrices. \ddot{x} , \dot{x} and x denote the acceleration, velocity and position of the end-effector of the manipulator, respectively. Through inverse kinematics and (2), the environmental dynamics in (3) can be transformed to joint space to describe the environment dynamics, such that

$$\Phi(\dot{q}, q) = \tau_{ext} \quad (4)$$

where $\Phi(\cdot)$ denotes the mapping function.

In this paper, a damping-stiffness admittance model [41] is used to describe this relationship, such as

$$C_d(\dot{x}_r - \dot{x}_d) + G_d(x_r - x_d) = -f_{ext} \quad (5)$$

where x_d and x_r denote the desired trajectory and modified desired trajectory in task space, respectively. C_d and G_d are coefficient matrices. According to (5), compliant motion x_r for the external force is generated based on the desired trajectory x_d and f_{ext} . In joint space, the admittance model (5) can be described as

$$C_d J(q)(\dot{q}_r - \dot{q}_d) + G_d(\varphi(q_r) - \varphi(q_d)) = -(J^T)^{-1} \tau_{ext} \quad (6)$$

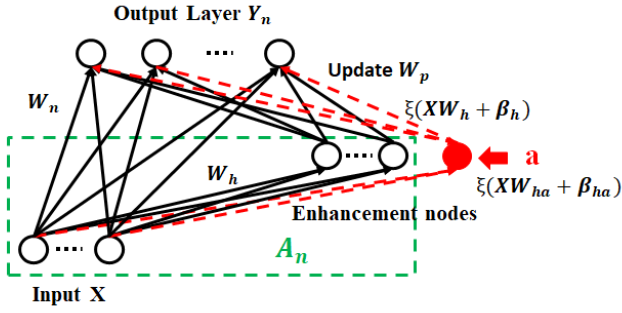


Fig. 1. The structure and updating algorithm for Functional-link neural network (modified from [19]).

where $\varphi(\cdot)$ denotes the robot kinematics function, q_r and q_d denote the modified desired trajectory and desired trajectory in joint space, respectively.

In admittance control schemes, the modified desired trajectory q_r is always assumed to be bounded, i.e., $|q_{ri}| < C_i$, ($i = 1, 2, \dots, n$), where C_i is a positive constant. However, during the process of robot-environment interaction, the modified trajectory q_r may violate the prescribed bound, that is, $|q_{ri}| > C_i$, $i = 1, 2, \dots, n$. To solve this problem, a saturation function is defined as

$$q_{ri} = \begin{cases} -\gamma_i(1 - e^{\frac{q_{ri} + \rho_i C_i}{\gamma_i}}) - \rho_i C_i, & q_{ri} < -\rho_i C_i \\ q_{ri}, & |q_{ri}| \leq \rho_i C_i \\ \gamma_i(1 - e^{\frac{\rho_i C_i - q_{ri}}{\gamma_i}}) + \rho_i C_i, & q_{ri} > \rho_i C_i \end{cases} \quad (7)$$

where $\gamma_i = (1 - \rho_i)C_i$, $0 < \rho_i < 1$ ($i = 1, 2, \dots, n$). Under this saturation function, the modified desired trajectory q_r can be guaranteed to stay within a constraint space.

B. RBFNN with PE Property

In this part, a brief introduction of RBFNN approximation is given. In [42], it is proven that the RBFNN can approximate any continuous function $f(\bullet)$ with a sufficient large node l , and optimal weight W^* , that is

$$f(Z_{in}) = W^{*T} S(Z_{in}) + \varepsilon(Z_{in}), \quad \forall Z_{in} \in \Omega_{Z_{in}} \quad (8)$$

where $Z_{in} \in \Omega_{Z_{in}} \subset R^v$ is the input vector of RBFNN with input dimension v . $\|\varepsilon(Z_{in})\| < \varepsilon^*$ denotes the approximation error ($\|\bullet\|$ is the Euclidean norm of vectors).

It should be noticed that the activation function of RBFNN is Gaussian function

$$s_i(Z_{in}) = \exp\left[-\frac{(Z_{in} - u_i^T)(Z_{in} - u_i)}{\eta_i^2}\right], \quad i = 1, \dots, l \quad (9)$$

where l is the number of the NN node; $u_i = [u_{i1}, u_{i2}, \dots, u_{iv}]$ ($i = 1, 2, \dots, l$) and η_i denotes the node center and the variance, respectively.

In practice, W^* is the ideal NN weight matrix and estimated by \hat{W} , and the function can be approximated by

$$\hat{f}(Z_{in}) = \hat{W}^T S(Z_{in}) \quad (10)$$

where $\hat{f}(Z_{in})$ is the estimate of $f(Z_{in})$. The approximation error of the weight can be defined as $\hat{W} = \hat{W} - W^*$.

Definition 1: [32] Consider the lattice of RBFNN is constructed in (8), we can select

$$\varrho \geq 2h = \min_{i \neq j} \|u_i - u_j\| > 0 \quad (11)$$

When the periodic input trajectory $Z'(t)$ is upon on the lattice, there will always be a regression subvector $S_\xi(Z')$ in the ϱ -neighborhood of $Z'(t)$, such that

$$S_\xi(Z') = [s_1(Z'_1), \dots, s_{l_\xi}(Z'_{l_\xi})] \quad (12)$$

with different centers $u_{i1}, \dots, u_{il_\xi}$. Because the value of the regression vector becomes very small as the trajectory moves away from the centers, we can choose

$$2h \leq \varrho \leq \varrho' \quad (13)$$

with the distance between Z' ($i = 1, 2, \dots, l_\xi$) and NN centers satisfying $\|Z'_i - u_{ij}\| < \varrho'$, we can obtain $s_i(Z') > \varrho''$, where ϱ'' is a small positive constant.

Lemma 1: [32] Assume that the periodic trajectory $Z'(t)$ is within a compact set $\Omega \in R^v$, and its derivative form $\dot{Z}'(t)$ is bounded within Ω . If the centers of RBFNN could be placed on a sufficient large enough lattice to cover the compact set Ω , the $S_\xi(Z')$ defined in (12) and (13) will satisfy the persistent excitation.

C. RVFLNN with Dynamic Updating Algorithm

The RVFLNN has a good ability of quick modeling for nonlinear systems with a simple structure, as shown in Fig. 1, where $\xi(\cdot)$ represents a mapping function for input X , and W_h, β_h are randomly generated, Y_n is the output matrix with $Y_n = A_n W_n$. In [19], an improvement version and a dynamic updating algorithm for RVFLNN is proposed to calculate weights for new added enhancement nodes. The model is modified and illustrated in Fig. 1, where the pattern matrix $A_n := [X|\xi(XW_h + \beta_h)]$ contains information of both input and enhancement nodes, and a is a new enhancement node added into the network. In this part, the dynamic updating algorithm for calculating the weights based on the matrix pseudo-inverse will be introduced with an added enhancement node. First, we define $A_p = [A_n|a]$, and the pseudoinverse of the matrix A_p is

$$A_p^+ = \begin{bmatrix} A_n^+ - db^T \\ b^T \end{bmatrix} \quad (14)$$

where $d = A_n^+ a$, and A_n^+ denotes pseudoinverse of the matrix A_n .

$$b^T = \begin{cases} (I + d^T d) d^T A_n^+ & \text{if } c = 0 \\ c^+ & \text{if } c \neq 0 \end{cases} \quad (15)$$

where $c = a - A_n d$. Then, the weights W_n and W_p can be computed by

$$\begin{aligned} W_n &= (\lambda I + A_n^T A_n)^{-1} A_n^T Y_n \\ W_p &= \begin{bmatrix} W_n - db^T Y_n \\ b^T Y_n \end{bmatrix} \end{aligned} \quad (16)$$

where W_n and W_p denote the weights before and after new enhancement nodes are added into the network. As can be seen from the above equations, after adding new enhancement nodes, we only need to calculate the weight of the newly added

nodes and do not need to recalculate the weight of the whole network.

Remark 1: As presented in [30], it has been noted that there is no restriction on the the mapping feature function $\xi(\cdot)$ from the input vector to the enhancement vector. In addition, this flatted neural network structure and the corresponding incremental learning algorithm can be applied in other neural network structures, such as support vector machine or RBF. Therefore, we choose the Gaussian function as the mapping function.

D. Actuator Deadzone

The deadzone nonlinearity is to describe the insensitivity of a system to small signals, as shown in Fig. 2.

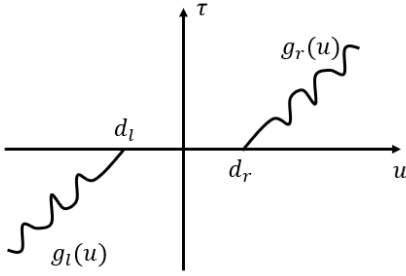


Fig. 2. The deadzone nonlinearity (modified from [36]).

The deadzone nonlinearity can be described as [36]

$$D(\tau) = \begin{cases} g_r(u - d_r), & \tau \geq d_r \\ 0, & d_l < \tau < d_r \\ g_l(u - d_l), & \tau \leq d_l \end{cases} \quad (17)$$

where u is the actuator input; $g_r(\cdot)$ and $g_l(\cdot)$ are the unknown functions; d_r and d_r are unknown parameters of the functions.

III. CONTROL DESIGN

A. Inverse Dynamics Method

In this section, a novel inverse-dynamics-based observer [17] with NNs is developed, and the revised architecture is shown in Fig. 3. The dynamics of the robotic arm in (1) can be Linearized as

$$\begin{aligned} \tau + J^T f_{ext} &= M_q(q)\ddot{q} + C_q(q, \dot{q})\dot{q} + G_q(q) \\ &= \Psi(\ddot{q}, \dot{q}, q)\Pi \\ &= \tau_{free} \end{aligned} \quad (18)$$

where τ_{free} is the torque to make the robotic arm move in free motion. The right side of the (18) is parameterized. Ψ is the regression matrix and Π is the parameter vector of the robotic arm. In this case, the estimation accuracy may be affected by inaccurate models or modelling errors. Traditional idea is to find a least squares solution of Π , and use it to estimate the contact force f_{ext} , such that

$$\begin{aligned} \hat{f}_{ext} &= J^{-T}(\Psi(\ddot{q}, \dot{q}, q)\hat{\Pi} - \tau) \\ &= J^{-T}(\hat{\tau}_{free} - \tau) \end{aligned} \quad (19)$$

where $\hat{\Pi}$ is estimate of the parameter vector. It is not difficult to find that the accuracy of the model directly affects the solution

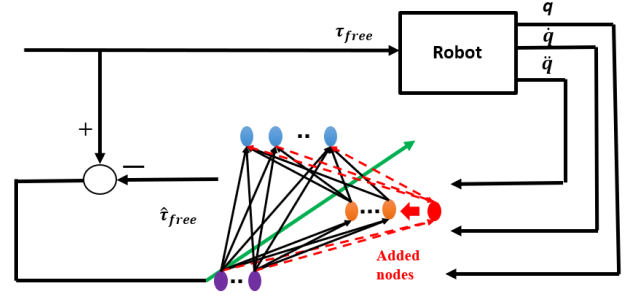


Fig. 3. The NN-based observer under training (modified from [17]).

of the least squares. It is difficult to implement the inverse-dynamics-based observer if the robotic arm is not accurately modelled or the modelling error is relatively large.

To solve this problem, A flatted network structure with RBF is developed to estimate the model parameters. Compared with traditional multilayer feedforward neural networks employed in [17], this NN structure incorporating incremental learning algorithm can guarantee the approximation accuracy without remodelling the NN structure, when the approximation accuracy is not satisfactory. As seen in (16), the advantage of incremental learning method is that the network only needs to train the weights of new added enhancement nodes without calculating the weights of the previous trained network. As depicted in Fig. 3, the inputs of the neural network under training are vectors of joint acceleration, velocity and joint position. After completing the training process, the trained NN model can be used to estimate the torque τ_{free} and calculate the external force from (19).

B. RBFNN Learning with Dynamic Framework

In this section, inspired the broad learning system, a dynamic RBFNN learning framework that combines node expansion and deterministic learning theory will be developed. As presented in Fig. 4, the NN nodes with centers far away from the input trajectory will be discarded, and then the number of NN nodes will be decreased. Then newly added nodes with centers close to input trajectory will be properly placed, and the number of NN nodes will increase. Considering PE conditions and *Lemma 1*, the satisfied parameter estimation could not be achieved well when the input of RBFNN is far from the centers of the Gaussian function, which will bring a low impact on the estimation performance. In this case, an extra NN node will be added to incorporate the new information. Here, the information of the added NN node is defined as $[u_{new}, \eta_{new}, W_{new}]$, where $u_{new}, \eta_{new}, W_{new}$ denote the center, variance and the weight of the added NN node respectively. Suppose that the distance between centers of the nearest m NN nodes and the corresponding input is $d_p = [p_1, p_2, \dots, p_m]$. Then, the parameters of the added node can be defined as

$$\begin{aligned} u_{new} &= \bar{p} + \gamma \|Z_\xi - \bar{p}\| \\ \eta_{new} &= \eta_i \end{aligned} \quad (20)$$

where Z_ξ is the current point of Z_{in} , the initial weight of added NN node is $W_{new} = 0$, \bar{p} denotes the average distance

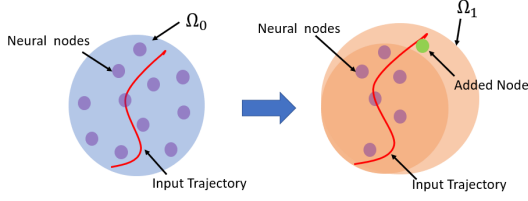


Fig. 4. The traditional RBFNN learning principle to enhancement of RBFNN learning (modified from [43]).

of d_p , such as

$$\bar{p} = \frac{\sum_{k=1}^m p_i}{m}, \quad i = 1, 2, \dots, m \quad (21)$$

After that, centers vector of NN nodes can be rewritten as

$$u_{t+1} = \begin{cases} [u_t, u_{new}], & \text{if } \bar{p} > \Xi \\ u_t, & \text{otherwise} \end{cases} \quad (22)$$

where Ξ is the predefined threshold. After new enhancement nodes are added into networks, the weight vector can be rewritten as

$$W_{(\cdot, n)} = \begin{bmatrix} W_{(\cdot)} \\ W_{new} \end{bmatrix} \quad (23)$$

where $W_{(\cdot, n)}$ is the weight vector after new enhancement nodes added into networks.

C. RBFNN Controller

In this part, a RBFNN-based controller is designed to guarantee the control performance. The tracking errors are defined as

$$\begin{aligned} e_1 &= q_r - q \\ \alpha &= \dot{q}_r + K_1 e_1 \\ e_2 &= \dot{e}_1 + K_1 e_1 \end{aligned} \quad (24)$$

where $K_1 = \{k_{11}, k_{12}, \dots, k_{1n}\}$ with k_{1i} being positive constant.

Considering the actuator deadzone, the robot dynamics (1) can be written as

$$M_q(q)\ddot{q} + C_q(q, \dot{q})\dot{q} + G_q(q) = D(\tau) + \tau_{ext} \quad (25)$$

where $D(\cdot)$ is defined in (17).

Design the control torque

$$\tau = \hat{M}_q \dot{\alpha} + \hat{C}_q \alpha + \hat{G}_q + K_2 e_2 + \Delta \hat{\tau} - \hat{\tau}_{ext} \quad (26)$$

where $\Delta \hat{\tau}$ is the estimates of $\Delta \tau = \tau - D(\tau)$, \hat{M}_q , \hat{C}_q and \hat{G}_q are the estimates of M_q , C_q and G_q , $\hat{\tau}_{ext}$ is the estimate of τ_{ext} ; K_2 is the control gain.

The updating law of RBFNN is

$$\begin{aligned} \dot{\hat{W}}_M &= \Theta_M^{-1} (S(X_M) \alpha e_2 - \sigma_M \hat{W}_M) \\ \dot{\hat{W}}_C &= \Theta_C^{-1} (S(X_C) \alpha e_2 - \sigma_C \hat{W}_C) \\ \dot{\hat{W}}_G &= \Theta_G^{-1} (S(X_G) e_2 - \sigma_G \hat{W}_G) \\ \dot{\hat{W}}_\tau &= -\Theta_\tau^{-1} (S(X_\tau) e_2 + \sigma_\tau \hat{W}_\tau) \end{aligned} \quad (27)$$

where $\Theta_{(\cdot)}$ are positive definite matrices; $\sigma_{(\cdot)}$ are positive constants for disturbance [32].

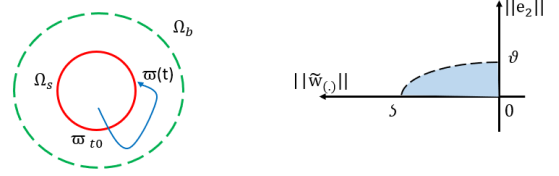


Fig. 5. Description of the UUB and the definition in (38).

Substituting (26) into (25) yields

$$\begin{aligned} -M_q \dot{e}_2 &= (\hat{M}_q - M_q) \dot{\alpha} + (\hat{C}_q - C_q) \alpha + (\hat{G}_q - G_q) \\ &\quad + C_q e_2 - \Delta \hat{\tau} + K_2 e_2 + \Delta \tau_e \end{aligned} \quad (28)$$

where $\Delta \tau_e = \tau_{ext} - \hat{\tau}_{ext}$.

Theorem 1: Considering the system dynamics (1), error signals (24), NN updating law (27) and Lyapunov function (32), the proposed control scheme can ensure that e_2 , $\|\hat{W}_{(\cdot)}\|$ are UUB. Since $\|\hat{W}_{(\cdot)}\|$ is bounded, $\|\hat{W}_{(\cdot)}\| = \|\hat{W}_{(\cdot)}^* + \hat{W}_{(\cdot)}\|$ will be bounded. According to the definition in (24), we can obtain e_1 , α , q are bounded.

Remark 2: In [32], it has been proven that almost all periodic trajectories satisfy the (partial) PE condition. The approximation of uncertainties can be achieved with the localized RBF neural networks ($\hat{W}^T S(Z_{in})$) when the tracking convergence is obtained. In a possible large lattice on which the RBF neural network is constructed, the input trajectory could not search all the every neural node. Therefore, the NN learning will only occur in the localized area near the input trajectory of the neural network. For those neurons with centers close to the input trajectory, their weights can converge to a small neighborhoods of a set of the optimal weights. For those neurons far from the input trajectory, their weight will updates slightly, with little change. Since the centers are far from the input trajectory, the activation function $S(Z_{in})$ will be small and the weights remain small according to the updating law (27), which means little learning experience can be obtained. Therefore, we can present the restriction between the input orbit and centers to make the RBF neural network have the learned knowledge, that is

$$dis(q_d, u) < \Upsilon \Rightarrow |\hat{W}^T S(Z_{in}) - f(x)| < \varepsilon \quad (29)$$

where $f(\cdot)$ denotes the unknown function, and Υ is a threshold specified by the designer. In this case, the unknown function is approximated by

$$\begin{aligned} f(x) &= W^{*T} S(Z_{in}) + \varepsilon(Z_{in}) \\ &= \hat{W}_\xi^T S(Z_{in}) + \varepsilon'(Z_{in}) \end{aligned} \quad (30)$$

where the \hat{W}_ξ is the subvector of \hat{W} , and ε' is the approximation error. \hat{W}_ξ corresponds to the NN nodes with their centers close to the input trajectory.

Considering that these neurons with centers far from the input orbit contribute little and bring extra computational burden to the system, in this paper, we will discard the NN nodes with centers far from the input trajectory according to the following principle

$$S(Z_{in}) < \lambda, \quad Z_{in} \in \mathbb{R}^v \quad (31)$$

where λ is a predefined constant.

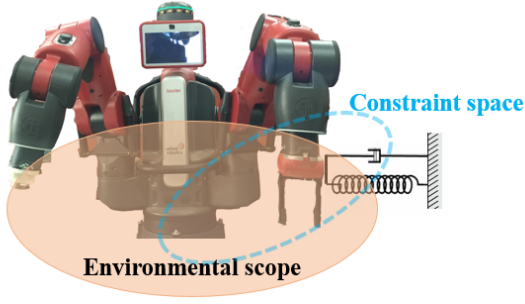


Fig. 6. Illustration of the experimental setup.

IV. EXPERIMENT RESULTS

In this section, experiments are conducted to verify the effectiveness of our proposed method. The Baxter robot is a humanoid two-arm robot, each arm consists of seven joints: two shoulder joints s_0, s_1 , two elbow joints e_0, e_1 and three wrist joints w_0, w_1, w_2 . The model dynamic is introduced in [44]. As shown in Fig. 6, the left arm of the Baxter robot is interacting with the external force form the environment, and the dynamics of the environment is modelled as the damping-stiffness system. The motion range of the robotic arm is specified within the constraint space.

A. Test of Adaptive RBFNN Controller

Firstly, the group of experiments aims to test the effectiveness of the RBFNN-based controller in the presence of actuator deadzone on joint s_0 . The upper and lower bounds of the deadzone input are defined as $d_r = 0.49$ and $d_l = -0.49$. The RBFNNs are divided into two groups, one for approximating the unknown dynamics of robot system with $[q, \dot{q}, \alpha, \dot{\alpha}, e_2] \in R^{5n}$, the other for dealing with the problem of input deazone, with $[\tau, q, \dot{q}, e_2] \in R^{4n}$. The center value of Guassian function of neural networks nodes should be set within the range of the upper and lower bound of the robot motion and input limits, in $[-1.5, 1.5] \times [-1.3, 1.3] \times [-1.3, 1.3] \times [-1.3, 1.3] \times [-1, 1] \cup [-10, 10] \times [-1.3, 1.3] \times [-1.5, 1.5] \times [-1, 1]$. The gains in updating law are selected as $\Theta_{\tau n} = \text{diag}\{0.125\}$, $\Theta_M = \text{diag}\{0.3\}$, $\Theta_C = \text{diag}\{0.3\}$, $\Theta_G = \text{diag}\{0.3\}$. The control parameters are selected as $K_1 = \text{diag}\{12\}$, $K_2 = \text{diag}\{6\}$. The desired trajectory is specified as $q_d = [\sin(5t); -1; 1.19; 1.94; -0.67; 1.03; -0.5]$.

The results are depicted in Figs. 7-12. As the input signal enters the deadzone space, the actuator could only provide little energy which cannot ensure the control performance, as shown in Fig. 10. In the Fig. 12, we can find that the deadzone condition (red solid line) is obvious at 10.5s and continued to 12.3s and difference between D_τ and control torque τ will degrade the control performance as shown in Fig. 10. With the NN compensation, the tracking performance has been improved and the tracking error (red dotted line) with NN compensation is smaller than the controller (blue solid line) without compensation, as depicted in Figs. 9 and 11. Based on the forward kinematics of Baxter robot is given in [45], the tracking performance of end-effector with NN compensation is shown in Figs. 7-8. As shown in Fig. 7, the actual trajectory of the end-effector can follow the desired trajectory effectively, and the tracking error is less than 0.04. As shown in Fig. 12,

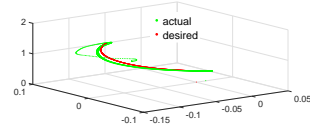


Fig. 7. The tracking performance of the end-effector.

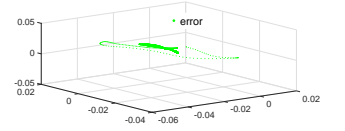


Fig. 8. The tracking error of the end-effector.

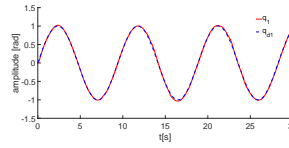


Fig. 9. The tracking performance of joint s_0 with NN compensation.

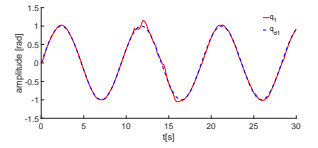


Fig. 10. The tracking performance of joint s_0 without NN compensation.

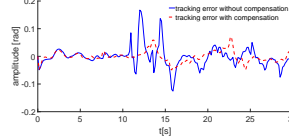


Fig. 11. The tracking errors of joint s_0 with/without NN compensation.

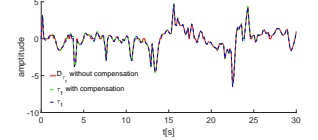


Fig. 12. The control inputs of joint s_0 .

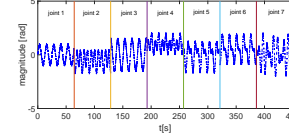


Fig. 13. The free-motion training trajectory.

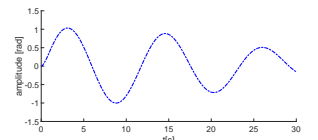


Fig. 14. The validation trajectory.

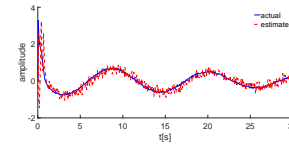


Fig. 15. The validation performance with proposed observer.

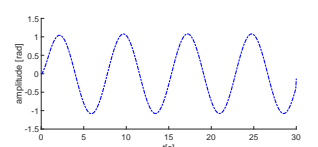


Fig. 16. The test trajectory.

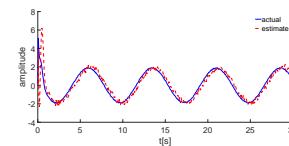


Fig. 17. The test performance with proposed observer.

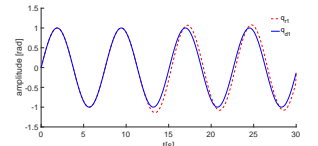


Fig. 18. The trajectory adaptation of joint s_0 .

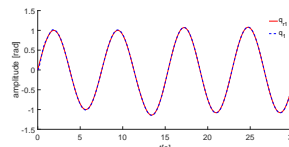


Fig. 19. The trajectory tracking of joint s_0 .

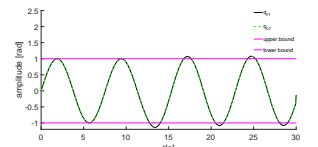


Fig. 20. The performance of trajectory constraint.

the input signal (green dotted line) with NN compensation is closer to the desired control input (blue solid line). The results of this group of experiments show the effectiveness of our proposed method.

B. Test of Admittance Control

This group of experiments is to test the performance of force estimation and trajectory modification based on admittance

control. In Fig. 6, the environment model is defined as: $-\tau_{ext} = 0.01\ddot{q} + 0.15\dot{q} + 1.1q$. The damping and stiffness matrices are defined as $C_E = \text{diag}\{3\}$, $G_E = \text{diag}\{5\}$. The parameters in saturation function are set as $\rho_i = 0.95$, $C_i = 1.1$, $\gamma_i = 0.05$.

Before testing the admittance control, the flatted function-link networks should be trained to obtain the robot dynamics parameters. During the training phase, seven joints of Baxter robot must be excited and controlled to move in free motion in the workspace. At the beginning, the trajectory of the robotic arm are commanded to flow a desired trajectory under the proportional-derivative (PD) controller. The motion of the robot arm should be set between the upper and lower bound of the motion range, in $[-1.7, 1.7] \times [-2.14, 1.05] \times [-3.05, 3.05] \times [-0.15, 2.61] \times [-3.05, 3.05] \times [-1.57, 2.09] \times [-3.05, 3.05]$. Then, the data of $q, \dot{q}, \ddot{q}, \tau$ are recorded for off-line training, as shown in Fig. 3. The training trajectories for each joint are depicted as shown in Fig. 13. The training time lasts for 450s, and each joint has different amplitudes and periods, as shown in Fig. 13. In the step of training, the arm of Baxter robot is required to be followed and excited for several times. After that, the proposed observer is validated and tested by similar trajectories with different amplitudes and periods, and the results are shown in Figs. 14-17. As shown in Figs. 14-17, the estimate can follow the external torque well, and the overall results are satisfactory.

The results of trajectory adaptation are presented in Figs. 18-20. The effect of external torque is about from 12s to 30s. As we can see from Fig. 18, when external force is applied to the manipulator, the desired trajectory (blue solid line) will be modified (red dotted line) to be compliant to the external force. The tracking performance after the trajectory modification is shown in Fig. 19. The external torque is estimated by the proposed observer in Fig. 17. The consuming time for training the neural network is 5.088s. The total number of NN nodes employed in the flatted NN structure is 1000. By using the saturation function, the robot motion is limited between the upper and lower bound predefined by the designer, which ensure the robot work within the prescribed joint space, as shown in Fig. 20. As we can see from these figures, the overall result is satisfactory, which shows the effectiveness of our proposed method.

C. Test of RBFNN learning with Dynamic Framework

This group of experiments aims to test the function approximation capabilities of RBFNN learning with dynamic updating algorithm. The RBFNN is employed to approximate the non-linear function $f(q, \dot{q}, \alpha, \dot{\alpha}) = M_q(q)\dot{\alpha} + C_q(q, \dot{q})\alpha + G_q(q)$, where α is defined in (24). The functional approximation performance of the our implemented neural network is depicted as shown in Figs. 21-26. The RBF network contains 512 nodes and parameters in updating law are set: $\Theta_{(\cdot)} = 0.03$, $\sigma = 0.035$. The parameter of discarding nodes defined in (29) is $\Upsilon = 0.05$. As shown in Fig. 22, the weight of NN nodes with their centers close to the input trajectory are activated well and the weight of NN nodes with their centers far away from the input trajectory are updating slightly and remains very small, that means, little learning experience can be obtained. In this case, we discard the NN nodes with their centers far away from the input trajectory according to the principle in (29), and the number of the discarded NN nodes is 208. After discarding

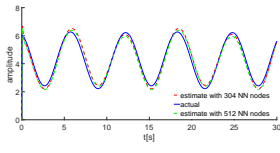


Fig. 21. The function approximation with 512 NN nodes and 304 NN nodes.

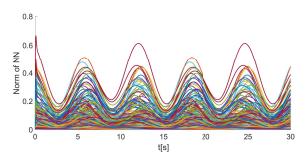


Fig. 22. The convergence of NN weight with 512 NN nodes.

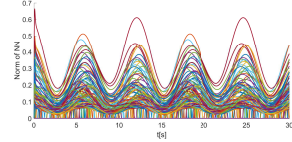


Fig. 23. The convergence of NN weight with 304 NN nodes.

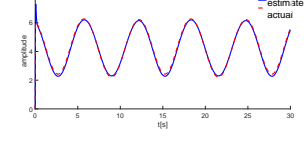


Fig. 24. The function approximation with dynamic RBFNN learning framework.

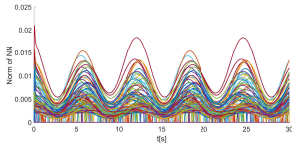


Fig. 25. The convergence of NN weight with 304 NN nodes under dynamic RBFNN learning framework.

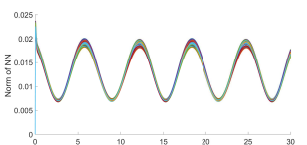


Fig. 26. The convergence of NN weight of 100 new NN nodes.

these NN nodes, the function approximation performance (red dotted line) is shown in Fig. 21 and the weight convergence is shown in Fig. 23. Comparing the approximation performance before (green dotted line) and after (red dotted line) discarding nodes as shown in Fig. 21, we can see that the performance of neural network function approximation is affected little after discarding these nodes, but the computational burden of the system has been reduced in half. From the Fig. 21, it is seen that the function approximation performance is not very satisfactory. To make the neural network have better approximation ability, 100 new nodes are added into system with their centers next to the input trajectory, and the approximation results are depicted in Fig. 24. The value of weight is updating according to the law in (27), and the convergence of the weight of old NN nodes and new NN nodes are shown in Figs. 25 and 26, respectively. By comparing the Fig. 21 with Fig. 24, we can see that the approximation performance (blue solid line in Fig. 24) of implemented neural network with dynamic framework could follow the approximated non-linear function's dynamics (red dotted line) better, which can verify the effectiveness of our proposed method.

V. CONCLUSION

In this paper, a neural-learning-based sensorless control scheme in the presence of input deadzone is presented for robotic arm to interact with the unknown environment. An inverse-dynamics-based force observer integrating the flatted network with RBF is developed to estimate the external force from the environment. A dynamic RBFNN learning framework is developed to approximate the uncertainties of the system. This novel NN adaptive controller is presented in Section III. The proposed control scheme is validated through experiments on the Baxter robot in Section IV.

VI. APPENDIX

Consider the following Lyapunov candidate

$$V = \frac{1}{2}e_2^T M_q e_2 + \frac{1}{2}tr(\tilde{W}_\tau^T \Theta_\tau \tilde{W}_\tau) + \frac{1}{2}tr(\tilde{W}_M^T \Theta_M \tilde{W}_M) + \frac{1}{2}tr(\tilde{W}_C^T \Theta_C \tilde{W}_C) + \frac{1}{2}tr(\tilde{W}_G^T \Theta_G \tilde{W}_G) \quad (32)$$

The derivative of (32) is

$$\dot{V} = e_2^T M_q \dot{e}_2 + \frac{1}{2}e_2^T \dot{M}_q e_2 + tr(\tilde{W}_\tau^T \Theta_\tau \dot{\tilde{W}}_\tau) + tr(\tilde{W}_M^T \Theta_M \dot{\tilde{W}}_M) + tr(\tilde{W}_C^T \Theta_C \dot{\tilde{W}}_C) + tr(\tilde{W}_G^T \Theta_G \dot{\tilde{W}}_G) \quad (33)$$

Substituting (31) into (33), we have

$$\dot{V} = e_2^T (-\tilde{M}\dot{\alpha} - \tilde{C}\dot{\alpha} - \tilde{G} + \Delta\hat{\tau}) - e_2^T K_2 e_2 - e_2^T \Delta\tau_e + tr(\tilde{W}_\tau^T \Theta_\tau \dot{\tilde{W}}_\tau) + tr(\tilde{W}_M^T \Theta_M \dot{\tilde{W}}_M) + tr(\tilde{W}_C^T \Theta_C \dot{\tilde{W}}_C) + tr(\tilde{W}_G^T \Theta_G \dot{\tilde{W}}_G) \quad (34)$$

Substituting the NN updating law (27) into (34), yields

$$\begin{aligned} \dot{V} &= -e_2^T (K_2 e_2 + \Delta\tau_e - \Delta\hat{\tau}) - e_2^T \tilde{M}\dot{\alpha} \\ &\quad - e_2^T \tilde{C}\dot{\alpha} - e_2^T \tilde{G} + tr(\tilde{W}_M^T \Theta_M \dot{\tilde{W}}_M) \\ &\quad + tr(\tilde{W}_C^T \Theta_C \dot{\tilde{W}}_C) + tr(\tilde{W}_G^T \Theta_G \dot{\tilde{W}}_G) \\ &= -e_2^T (K_2 e_2 + \Delta\tau_e - \Delta\hat{\tau}) - e_2^T \tilde{M}\dot{\alpha} \\ &\quad - e_2^T \tilde{C}\dot{\alpha} - e_2^T \tilde{G} + tr[\tilde{W}_\tau^T (S(X_\tau)e_2 - \sigma_\tau \hat{W}_\tau)] \\ &\quad + tr[\tilde{W}_M^T (S(X_M)\dot{\alpha}e_2 - \sigma_M \hat{W}_M)] \\ &\quad + tr[\tilde{W}_C^T (S(X_C)\alpha e_2 - \sigma_C \hat{W}_C)] \\ &\quad + tr[\tilde{W}_G^T (S(X_G)e_2 - \sigma_G \hat{W}_G)] \\ &= -e_2^T K_2 e_2 - e_2^T \Delta\tau_e - \sigma_M tr(\tilde{W}_M^T \tilde{W}_M) \\ &\quad - \sigma_M tr(\tilde{W}_M^T W_M^*) - \sigma_C tr(\tilde{W}_C^T \tilde{W}_C) \\ &\quad - \sigma_C tr(\tilde{W}_C^T W_C^*) - \sigma_G tr(\tilde{W}_G^T \tilde{W}_G) \\ &\quad - \sigma_G tr(\tilde{W}_G^T W_G^*) - \sigma_\tau tr(\tilde{W}_\tau^T \tilde{W}_\tau) \\ &\quad - \sigma_\tau tr(\tilde{W}_\tau^T W_\tau^*) \end{aligned} \quad (35)$$

Using the Youngs inequality [46], and the derivative of V is

$$\dot{V} \leq -e_2^T K_2 e_2 + \frac{1}{2}e_2^T e_2 - \frac{1}{2}\sigma_\tau tr(\tilde{W}_\tau^T \tilde{W}_\tau) - \frac{1}{2}\sigma_M tr(\tilde{W}_M^T \tilde{W}_M) - \frac{1}{2}\sigma_C tr(\tilde{W}_C^T \tilde{W}_C) - \frac{1}{2}\sigma_G tr(\tilde{W}_G^T \tilde{W}_G) + \Delta \quad (36)$$

where $\Delta = \frac{1}{2}\Delta\tau_e^T \Delta\tau_e + \frac{1}{2}\sigma_M tr(W_M^{*T} W_M^*) + \frac{1}{2}\sigma_C tr(W_C^{*T} W_C^*) + \frac{1}{2}\sigma_G tr(W_G^{*T} W_G^*) + \frac{1}{2}\sigma_\tau tr(W_\tau^{*T} W_\tau^*)$.

Let us define ϖ comprised of $e_2, \tilde{W}_M, \tilde{W}_C, \tilde{W}_G$, and the (36) can be rewritten as $\dot{V} \leq -\kappa\mu(\varpi) + \Delta$, where κ and Δ are positive constant. There is an invariant set Ω_s for $-\kappa\mu(\varpi) + \Delta > 0$, that is, the derivative of V is negative outside the set. We can define the set Ω_b with $V(\varpi)$ decreasing in $\dot{V} < 0$, after a period of time, the states ϖ will enter into Ω_s and will not escape afterwards. By uniformly ultimately bounded (UUB) stability, the state variable ϖ will converge to a bounded set.

The invariant set can be defined as

$$\Omega_s = \left\{ (||\tilde{W}_M||, ||\tilde{W}_C||, ||\tilde{W}_G||, ||\tilde{W}_\tau||, ||e_2||), \left| \frac{e_2^T (K_2 - \frac{1}{2}I) e_2}{\Delta} + \frac{\sigma_D ||\tilde{W}_D||^2}{\Delta} + \frac{\sigma_\Delta ||\tilde{W}_C||^2}{\Delta} + \frac{\sigma_G ||\tilde{W}_G||^2}{\Delta} + \frac{\sigma_\tau ||\tilde{W}_\tau||^2}{\Delta} \leq 1 \right. \right\} \quad (37)$$

As depicted in Fig. 5, the area of the set Ω_s is in the first quadrant passing through the points

$$\begin{aligned} \sigma_{(\cdot)} ||\tilde{W}_{(\cdot)}||^2 &= \Delta, & \tilde{W} &= \varsigma \\ e_2^T (K_2 - \frac{1}{2}I) e_2 &= \Delta, & e_2 &= \vartheta \end{aligned} \quad (38)$$

After new enhancement nodes are added into networks, we reconstruct a new Lyapunov candidate

$$V = \frac{1}{2}e_2^T M_q e_2 + \frac{1}{2}tr(\tilde{W}_{\tau n}^T \Theta_{\tau n} \tilde{W}_{\tau n}) + \frac{1}{2}tr(\tilde{W}_{MN}^T \Theta_{MN} \tilde{W}_{MN}) + \frac{1}{2}tr(\tilde{W}_{CN}^T \Theta_{CN} \tilde{W}_{CN}) + \frac{1}{2}tr(\tilde{W}_{GN}^T \Theta_{GN} \tilde{W}_{GN}) \quad (39)$$

where $W_{(\cdot)}$ denotes the NN weight after adding new nodes into networks.

The derivative of (39) is

$$\dot{V} = e_2^T M_q \dot{e}_2 + \frac{1}{2}e_2^T \dot{M}_q e_2 + tr(\tilde{W}_{\tau n}^T \Theta_{\tau n} \dot{\tilde{W}}_{\tau n}) + tr(\tilde{W}_{MN}^T \Theta_{MN} \dot{\tilde{W}}_{MN}) + tr(\tilde{W}_{CN}^T \Theta_{CN} \dot{\tilde{W}}_{CN}) + tr(\tilde{W}_{GN}^T \Theta_{GN} \dot{\tilde{W}}_{GN}) \quad (40)$$

Substituting (28) into (40) and noting that

$$\begin{aligned} \dot{V} &= e_2^T (-\tilde{M}\dot{\alpha} - \tilde{C}\dot{\alpha} - \tilde{G} + \Delta\hat{\tau}) - e_2^T K_2 e_2 \\ &\quad - e_2^T \Delta\tau_e + tr \left(\begin{bmatrix} \tilde{W}_\tau \\ \tilde{W}_{\tau new} \end{bmatrix}^T \Theta_{\tau n} \begin{bmatrix} \dot{\tilde{W}}_\tau \\ \dot{\tilde{W}}_{\tau new} \end{bmatrix} \right) \\ &\quad + tr \left(\begin{bmatrix} \tilde{W}_M \\ \tilde{W}_{M new} \end{bmatrix}^T \Theta_{MN} \begin{bmatrix} \dot{\tilde{W}}_M \\ \dot{\tilde{W}}_{M new} \end{bmatrix} \right) \\ &\quad + tr \left(\begin{bmatrix} \tilde{W}_C \\ \tilde{W}_{C new} \end{bmatrix}^T \Theta_{CN} \begin{bmatrix} \dot{\tilde{W}}_C \\ \dot{\tilde{W}}_{C new} \end{bmatrix} \right) \\ &\quad + tr \left(\begin{bmatrix} \tilde{W}_G \\ \tilde{W}_{G new} \end{bmatrix}^T \Theta_{GN} \begin{bmatrix} \dot{\tilde{W}}_G \\ \dot{\tilde{W}}_{G new} \end{bmatrix} \right) \end{aligned} \quad (41)$$

where $W_{(\cdot new)}$ denotes the weight vector of new added nodes.

The updating law is designed as

$$\begin{aligned} \dot{\tilde{W}}_{MN} &= \Theta_{MN}^{-1} (S(X_{MN})\dot{\alpha}e_2 - \sigma_{MN} \hat{W}_{MN}) \\ \dot{\tilde{W}}_{CN} &= \Theta_{CN}^{-1} (S(X_{CN})\alpha e_2 - \sigma_{CN} \hat{W}_{CN}) \\ \dot{\tilde{W}}_{GN} &= \Theta_{GN}^{-1} (S(X_{GN})e_2 - \sigma_{GN} \hat{W}_{GN}) \\ \dot{\tilde{W}}_{\tau n} &= -\Theta_{\tau n}^{-1} (S(X_{\tau n})e_2 + \sigma_{\tau n} \hat{W}_{\tau n}) \end{aligned} \quad (42)$$

where $\Theta_{(\cdot)}$ are positive definite matrices, and $\sigma_{(\cdot)}$ are positive constant.

Substituting (42) into (41), we have

$$\begin{aligned} \dot{V} = & -e_2^T(K_2e_2 + \Delta\tau_e - \Delta\hat{\tau}) - e_2^T\tilde{M}\dot{\alpha} \\ & - e_2^T\tilde{C}\alpha - e_2^T\tilde{G} + tr(\tilde{W}_{MN}^T\Theta_{MN}\dot{W}_{MN}) \\ & + tr(\tilde{W}_{CN}^T\Theta_{CN}\dot{W}_{CN}) + tr(\tilde{W}_{GN}^T\Theta_{GN}\dot{W}_{GN}) \\ = & -e_2^TK_2e_2 - e_2^T\Delta\tau_e - \sigma_{MN}tr(\tilde{W}_{MN}^T\tilde{W}_{MN}) \\ & - \sigma_{MN}tr(\tilde{W}_{MN}^TW_{MN}^*) - \sigma_{CN}tr(\tilde{W}_{CN}^T\tilde{W}_{CN}) \\ & - \sigma_{CN}tr(\tilde{W}_{CN}^TW_{CN}^*) - \sigma_{GN}tr(\tilde{W}_{GN}^T\tilde{W}_{GN}) \\ & - \sigma_{GN}tr(\tilde{W}_{GN}^TW_{GN}^*) - \sigma_{\tau n}tr(\tilde{W}_{\tau n}^T\tilde{W}_{\tau n}) \\ & - \sigma_{\tau n}tr(\tilde{W}_{\tau n}^TW_{\tau n}^*) \end{aligned} \quad (43)$$

By using the Young's inequality in (43), yields

$$\begin{aligned} \dot{V} \leq & -e_2^TK_2e_2 + \frac{1}{2}e_2^Te_2 - \frac{1}{2}\sigma_{\tau n}tr(\tilde{W}_{\tau n}^T\tilde{W}_{\tau n}) \\ & - \frac{1}{2}\sigma_{MN}tr(\tilde{W}_{MN}^T\tilde{W}_{MN}) - \frac{1}{2}\sigma_{CN}tr(\tilde{W}_{CN}^T\tilde{W}_{CN}) \\ & - \frac{1}{2}\sigma_{GN}tr(\tilde{W}_{GN}^T\tilde{W}_{GN}) + \Delta_n \end{aligned} \quad (44)$$

where $\Delta_n = \frac{1}{2}\Delta\tau_e^T\Delta\tau_e + \frac{1}{2}\sigma_{MN}tr(W_{MN}^{*T}W_{MN}^*) + \frac{1}{2}\sigma_{CN}tr(W_{CN}^{*T}W_{CN}^*) + \frac{1}{2}\sigma_{GN}tr(W_{GN}^{*T}W_{GN}^*) + \frac{1}{2}\sigma_{\tau n}tr(W_{\tau n}^{*T}W_{\tau n}^*)$. We can find that the states after adding new enhancement nodes also satisfy uniformly ultimately bounded(UUB) stability. By using *Theorem 1*, we can prove that the states $\tilde{W}_{\tau n}, \tilde{W}_{MN}, \tilde{W}_{CN}, \tilde{W}_{GN}, e_2$ will converge to an invariant set.

REFERENCES

- [1] W. He, Z. Li, and C. P. Chen, "A survey of human-centered intelligent robots: issues and challenges," *IEEE/CAA Journal of Automatica Sinica*, vol. 4, no. 4, pp. 602–609, 2017.
- [2] Z. Li, S. S. Ge, and A. Ming, "Adaptive robust motion/force control of holonomic-constrained nonholonomic mobile manipulators," *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 37, no. 3, pp. 607–616, 2007.
- [3] J. J. Craig and M. H. Raibert, "A systematic method of hybrid position/force control of a manipulator," in *COMPSAC 79. Proceedings. Computer Software and The IEEE Computer Society's Third International Applications Conference, 1979*. IEEE, 1979, pp. 446–451.
- [4] S. Part, "Impedance control: An approach to manipulation," *Journal of dynamic systems, measurement, and control*, vol. 107, no. 17, 1985.
- [5] J. E. Colgate and N. Hogan, "Robust control of dynamically interacting systems," *International journal of Control*, vol. 48, no. 1, pp. 65–88, 1988.
- [6] M. T. Mason, "Compliance and force control for computer controlled manipulators," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 11, no. 6, pp. 418–432, 1981.
- [7] I. Ranatunga, F. L. Lewis, D. O. Popa, and S. M. Tousif, "Adaptive admittance control for human-robot interaction using model reference design and adaptive inverse filtering," *IEEE Transactions on Control Systems Technology*, vol. 25, no. 1, pp. 278–285, Jan 2017.
- [8] P. Marayong, G. D. Hager, and A. M. Okamura, "Control methods for guidance virtual fixtures in compliant human-machine interfaces," in *2008 IEEE/RSJ International Conference on Intelligent Robots and Systems*, Sep. 2008, pp. 1166–1172.
- [9] B. J. Waibel and H. Kazerooni, "Theory and experiments on the stability of robot compliance control," *IEEE Transactions on Robotics and Automation*, vol. 7, no. 1, pp. 95–104, 1991.
- [10] S. Eppinger and W. Seering, "Introduction to dynamic models for robot force control," *IEEE Control Systems Magazine*, vol. 7, no. 2, pp. 48–52, 1987.
- [11] C. Lee, S. Chan, and D. Mital, "A joint torque disturbance observer for robotic assembly," in *Proceedings of 36th Midwest Symposium on Circuits and Systems*. IEEE, 1993, pp. 1439–1442.
- [12] T. Murakami, R. Nakamura, F. Yu, and K. Ohnishi, "Force sensorless impedance control by disturbance observer," in *Conference Record of the Power Conversion Conference-Yokohama 1993*. IEEE, 1993, pp. 352–357.
- [13] P. Hacksel and S. Salcudean, "Estimation of environment forces and rigid-body velocities using observers," in *Proceedings of the 1994 IEEE International Conference on Robotics and Automation*. IEEE, 1994, pp. 931–936.
- [14] M. Capurso, M. M. G. Ardakani, R. Johansson, A. Robertsson, and P. Rocco, "Sensorless kinesthetic teaching of robotic manipulators assisted by observer-based force control," in *2017 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2017, pp. 945–950.
- [15] X. Liu, F. Zhao, S. S. Ge, Y. Wu, and X. Mei, "End-effector force estimation for flexible-joint robots with global friction approximation using neural networks," *IEEE Transactions on Industrial Informatics*, vol. 15, no. 3, pp. 1730–1741, March 2019.
- [16] V. Zahn, R. Maass, M. Dapper, and R. Eckmiller, "Learning friction estimation for sensorless force/position control in industrial manipulators," in *Proceedings 1999 IEEE International Conference on Robotics and Automation (Cat. No. 99CH36288C)*, vol. 4. IEEE, 1999, pp. 2780–2785.
- [17] A. C. Smith, F. Mobasser, and K. Hashtrudi-Zaad, "Neural-network-based contact force observers for haptic applications," *IEEE Transactions on Robotics*, vol. 22, no. 6, pp. 1163–1175, 2006.
- [18] Y.-H. Pao, G.-H. Park, and D. J. Sobajic, "Learning and generalization characteristics of the random vector functional-link net," *Neurocomputing*, vol. 6, no. 2, pp. 163–180, 1994.
- [19] C. L. P. Chen and J. Z. Wan, "A rapid learning and dynamic stepwise updating algorithm for flat neural networks and the application to time-series prediction," *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 29, no. 1, pp. 62–72, Feb 1999.
- [20] C.-E. Ren and C. P. Chen, "Sliding mode leader-following consensus controllers for second-order non-linear multi-agent systems," *IET Control Theory & Applications*, vol. 9, no. 10, pp. 1544–1552, 2015.
- [21] C. P. Chen, G.-X. Wen, Y.-J. Liu, and F.-Y. Wang, "Adaptive consensus control for a class of nonlinear multiagent time-delay systems using neural networks," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 25, no. 6, pp. 1217–1226, 2014.
- [22] Z. Li, S. Xiao, S. S. Ge, and H. Su, "Constrained multilegged robot system modeling and fuzzy control with uncertain kinematics and dynamics incorporating foot force optimization," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 46, no. 1, pp. 1–15, 2015.
- [23] Z. Liu, G. Lai, Y. Zhang, and C. L. P. Chen, "Adaptive neural output feedback control of output-constrained nonlinear systems with unknown output nonlinearity," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 26, no. 8, pp. 1789–1802, 2015.
- [24] D.-P. Li, D.-J. Li, Y.-J. Liu, S. Tong, and C. P. Chen, "Approximation-based adaptive neural tracking control of nonlinear mimo unknown time-varying delay systems with full state constraints," *IEEE transactions on cybernetics*, vol. 47, no. 10, pp. 3100–3109, 2017.
- [25] A. M. Smith, C. Yang, H. Ma, P. Culverhouse, A. Cangelosi, and E. Burdet, "Novel hybrid adaptive controller for manipulation in complex perturbation environments," *PloS one*, vol. 10, no. 6, 2015.
- [26] C. Yang, X. Wang, L. Cheng, and H. Ma, "Neural-learning-based telerobot control with guaranteed performance," *IEEE Transactions on Cybernetics*, vol. 47, no. 10, pp. 3148–3159, Oct 2017.
- [27] C. Yang, Y. Jiang, Z. Li, W. He, and C. Su, "Neural control of bimanual robots with guaranteed global stability and motion precision," *IEEE Transactions on Industrial Informatics*, vol. 13, no. 3, pp. 1162–1171, June 2017.
- [28] C. Yang, X. Wang, Z. Li, Y. Li, and C. Su, "Teleoperation control based on combination of wave variable and neural networks," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 47, no. 8, pp. 2125–2136, Aug 2017.
- [29] Z. Li, Y. Xia, and F. Sun, "Adaptive fuzzy control for multilateral cooperative teleoperation of multiple robotic manipulators under random network-induced delays," *IEEE Transactions on Fuzzy Systems*, vol. 22, no. 2, pp. 437–450, April 2014.
- [30] 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, Jan 2018.
- [31] C. L. P. Chen, Z. Liu, and S. Feng, "Universal approximation capability of broad learning system and its structural variations," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 30, no. 4, pp. 1191–1204, April 2019.
- [32] Cong Wang and D. J. Hill, "Learning from neural control," *IEEE Transactions on Neural Networks*, vol. 17, no. 1, pp. 130–146, Jan 2006.
- [33] F. L. Lewis, Woo Kam Tim, Li-Zin Wang, and Z. X. Li, "Deadzone compensation in motion control systems using adaptive fuzzy logic control," *IEEE Transactions on Control Systems Technology*, vol. 7, no. 6, pp. 731–742, Nov 1999.
- [34] P. L. Andrighetto, A. C. Valdiero, and D. Bavaresco, "Dead zone compensation in pneumatic servo systems," in *ABCM symposium series in mechatronics*, vol. 3, 2008, pp. 501–509.

- [35] R. R. Selmic and F. L. Lewis, "Deadzone compensation in motion control systems using neural networks," *IEEE Transactions on Automatic Control*, vol. 45, no. 4, pp. 602–613, April 2000.
- [36] W. He, A. O. David, Z. Yin, and C. Sun, "Neural network control of a robotic manipulator with input deadzone and output constraint," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 46, no. 6, pp. 759–770, June 2016.
- [37] S. I. Han and J. M. Lee, "Precise positioning of nonsmooth dynamic systems using fuzzy wavelet echo state networks and dynamic surface sliding mode control," *IEEE Transactions on Industrial Electronics*, vol. 60, no. 11, pp. 5124–5136, Nov 2013.
- [38] Y. Liu, L. Tang, S. Tong, and C. L. P. Chen, "Adaptive nn controller design for a class of nonlinear mimo discrete-time systems," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 26, no. 5, pp. 1007–1018, May 2015.
- [39] G. Peng, C. Yang, W. He, and C. L. P. Chen, "Force sensorless admittance control with neural learning for robots with actuator saturation," *IEEE Transactions on Industrial Electronics*, pp. 1–1, 2019.
- [40] B. Siciliano and O. Khatib, *Springer handbook of robotics*. Springer, 2016.
- [41] S. G. Shuzhi, C. C. Hang, and L. Woon, "Adaptive neural network control of robot manipulators in task space," *IEEE transactions on industrial electronics*, vol. 44, no. 6, pp. 746–752, 1997.
- [42] T. H. Lee and C. J. Harris, *Adaptive neural network control of robotic manipulators*. World Scientific, 1998, vol. 19.
- [43] 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*, pp. 1–1, 2019.
- [44] A. Smith, C. Yang, C. Li, H. Ma, and L. Zhao, "Development of a dynamics model for the baxter robot," in *2016 IEEE International Conference on Mechatronics and Automation*, Aug 2016, pp. 1244–1249.
- [45] Z. Ju, C. Yang, and H. Ma, "Kinematics modeling and experimental verification of baxter robot," in *Proceedings of the 33rd Chinese Control Conference*, July 2014, pp. 8518–8523.
- [46] W. H. Young, "On classes of summable functions and their fourier series," *Proceedings of the Royal Society of London. Series A, Containing Papers of a Mathematical and Physical Character*, vol. 87, no. 594, pp. 225–229, 1912.



Wei He (S'09-M'12-SM'16) received his B.Eng. and his M.Eng. degrees from College of Automation Science and Engineering, South China University of Technology (SCUT), China, in 2006 and 2008, respectively, and his PhD degree from Department of Electrical & Computer Engineering, the National University of Singapore (NUS), Singapore, in 2011. He is currently working as a full professor in School of Automation and Electrical Engineering, University of Science and Technology Beijing, Beijing, China. He has co-authored 3 books published in Springer and published over 100 international journal and conference papers. He was awarded a Newton Advanced Fellowship from the Royal Society, UK in 2017. He was a recipient of the IEEE SMC Society Andrew P. Sage Best Transactions Paper Award in 2017. He is serving the Chair of IEEE SMC Society Beijing Capital Region Chapter. He is serving as an Associate Editor of *IEEE Transactions on Neural Networks and Learning Systems*, *IEEE Transactions on Control Systems Technology*, *IEEE Transactions on Systems, Man, and Cybernetics: Systems, Assembly Automation*, *IEEE/CAA Journal of Automatica Sinica*, *Neurocomputing*, and an Editor of *Journal of Intelligent & Robotic Systems*. His current research interests include robotics, distributed parameter systems and intelligent control systems.



Guangzhu Peng received the B.Eng. degree in automation from Yangtze University, Jingzhou, China, in 2014, and the M.Eng. degree in pattern recognition and intelligent systems from the College of Automation Science and Engineering, South China University of Technology (SCUT), Guangzhou, China, in 2018. He is currently pursuing the Ph.D. degree in Computer Science with the Faculty of Science and Technology, University of Macau, Macau, China.

His current research interests include robotics, human-robot interaction, intelligent control, etc.



C. L. Philip Chen (S'88-M'88-SM'94-F'07) is the Chair Professor and Dean of the School of Computer Science and Engineering, South China University of Technology, China; on leave from Faculty of Science and Technology, University of Macau, Macau, China. The University of Macau's Engineering and Computer Science programs receiving Hong Kong Institute of Engineers' (HKIE) accreditation and Washington/Seoul Accord is his utmost contribution in engineering/computer science education for Macau as the former Dean of the Faculty.

His current research interests include systems, cybernetics, and computational intelligence. Dr. Chen was the IEEE SMC Society President from 2012 to 2013 and is now a Vice President of Chinese Association of Automation (CAA). He is a Fellow of IEEE, AAAS, IAPR, CAA, and HKIE. He is the editor-in-chief of the *IEEE Transactions on Cybernetics*, and an associate editor of several *IEEE Transactions*. He was a Program Evaluator of the Accreditation Board of Engineering and Technology Education (ABET) of the U.S. for computer engineering, electrical engineering, and software engineering programs. He received 2016 Outstanding Electrical and Computer Engineers award from his alma mater, Purdue University after he graduated from the University of Michigan, Ann Arbor, Michigan, USA.



Chenguang Yang (M'10-SM'16) received the B.Eng. degree in measurement and control from Northwestern Polytechnical University, Xian, China, in 2005, the Ph.D. degree in control engineering from the National University of Singapore, Singapore, in 2010 and postdoctoral training in human robotics from the Imperial College London, London, U.K. He has been awarded EU Marie Curie International Incoming Fellowship, UK EPSRC UKRI Innovation Fellowship, and the Best Paper Award of the *IEEE Transactions on Robotics* as well as over

ten conference Best Paper Awards. His research interest lies in human robot interaction and intelligent system design.