Adaptive dynamic programming enhanced admittance control for robots with environment interaction and actuator saturation

Hong Zhan¹ · Dianye Huang² · Chenguang Yang³

Received: 29 November 2019 / Accepted: 6 December 2020 / Published online: 2 February 2021
© The Author(s) 2021

Abstract

This paper focuses on the optimal tracking control problem for robot systems with environment interaction and actuator saturation. A control scheme combined with admittance adaptation and adaptive dynamic programming (ADP) is developed. The unknown environment is modelled as a linear system and admittance controller is derived to achieve compliant behaviour of the robot. In the ADP framework, the cost function is defined with non-quadratic form and the critic network is designed with radial basis function neural network which introduces to obtain an approximate optimal control of the Hamilton–Jacobi–Bellman equation, which guarantees the optimal trajectory tracking. The system stability is analysed by Lyapunov theorem and simulations demonstrate the effectiveness of the proposed strategy.

Keywords Adaptive dynamic programming · Admittance control · Robot-environment interaction · Actuator saturation · Optimal control · Neural network

1 Introduction

In recent decades, robots are widely applied in industrial automation, such as assembling robots, handling robots, welding robots. They can not only cooperate with human partners for certain work, but also can complete some tasks independently, or even replace human beings to work in some hazard environment with high temperature, pressure and radiation. However, in some practical applications, robots will unavoidably interact with the external environment, which will not only affect execution of the work, but also directly threaten safety of human partners and robots themselves. Consequently, interaction control between robot and the environment has become an important research topic.

It's noted that there are two main approaches applied in current research in robotics to ensure the compliant behaviour, i.e., hybrid position/force control proposed by Raibert and Craig (1981) and impedance control proposed by Hogan (1981). The former requires decomposition in position and force subspaces and control law switching during implementation process. Since the dynamic coupling between the robot and external environment is not considered, the accuracy of this approach is difficult to be guaranteed. Comparatively, the latter establishes the relationship between the robot and environment, and achieve compliant behaviour by adjusting mechanical impedance to a target value in case interaction occurs, which guarantees the interaction safety. Impedance control has two execution methods according to the controller causality, i.e., impedance control and admittance control. For impedance control system, the external force imposed by the environment can be obtained by desired trajectory and impedance model, while for admittance control system, the modified motion trajectory can be derived from the measured interaction force and the expected admittance model. Therefore, we adopt admittance control to deal with robot-environment interaction problem.

The interaction force and admittance model are significant parts for admittance control. If interaction between the robot and environment occurs, the interaction force can be measured by the force sensors which are mounted at the

✉ Chenguang Yang
cyang@ieee.org

¹ Key Lab. Autonomous Systems and Networked Control, Ministry of Education, South China University of Technology, Guangzhou 510640, China

² Department of Informatics, Technical University of Munich, 85748 München, Germany

³ Bristol Robotics Laboratory, University of the West of England, Bristol BS16 1QY, UK

end-effector of the robot. However, due to the complex environment, it's often very hard to obtain the desired admittance model which is critical for admittance control system. In addition, the fixed model can't satisfy requirements of all situations. Consequently, Braun et al. (2012) took human-robot cooperation as an example and proposed that it was essential to adopt variable admittance model to improve system efficiency. For variable admittance control, iterative learning has been studied to derive the admittance parameters to adapt to unknown environment in robot intelligent control field. To complete a wall-following task, Cohen and Flash (1991) proposed an impedance learning strategy with an associate network. Tsuji et al. (1996) introduced neural network into impedance control to tuning the model parameters. But iterative learning approach requires the robot to perform the same task repeatedly, which is not available in some practical application execution. So researchers have adopted adaptation methods to solve this problem such as Love and Book (2004), Uemura and Kawamura (2009), Stanisic and Fernandez (2012), Landi et al. (2017) and Yao et al. (2018).

Tracking control is a very important research topic in robot intelligent control area. In the current studies, a lot of control methods have been employed to robot systems. Cervantes and Alvarez-Ramirez (2001) and Parra-Vega et al. (2003) applied the classic proportional-integral-derivative(PID) control into the robot system with satisfied tracking performance. PID control is often used in the industrial field owing to the simple structure and good performance. But for complex systems, it is very difficult to choose appropriate PID parameters which normally depends on experiences of the operator. In recent years, neural network(NN) control has been investigated and applied to robot systems because of strong approximation property for unknown system (Yang et al. 2017). In Zhang et al. (2018), NN control was employed to improve the tracking performance of the robot system with uncertainties. In Yang et al. (2019), NN-based controller combined with admittance adaptation was proposed to tackle the robot-environment interaction problem. However, these control methods only deal with stabilization problem of the system without considering optimal control. Based on the optimal control theory, we expect to find a control strategy that enables the system to reach the target in an optimal manner. To achieve this goal, it is usually required to minimize the specified cost function by solving the Hamilton–Jacobi–Bellman (HJB) equation. The HJB equation for a nonlinear system is nonlinear partial differential, so its analytical solution is non-trivial to derive. Dynamic programming proposed by Bellman (1957) provides a useful method for solving HJB equation. However, since this method is studied based on backward numerical process, it will be affected by well-known curse of dimensionality with the increase of system dimension. To

overcome this problem, Werbos (1992) proposed adaptive dynamic programming (ADP) strategy using NN to approximate the cost function forward and then obtain the solution of HJB equation. During the past few years, great efforts have been made on ADP to deal with the control issues for nonlinear systems (Liu et al. 2014; Jiang and Jiang 2015), such as systems with dynamic uncertainties (Wang et al. 2018) and disturbances (Cui et al. 2017).

In practical control system, actuator saturation is a common phenomenon, which may affect the system performance, or even result in system instability. Therefore, it is essential and challenging to derive optimal control strategy for nonlinear systems with actuator saturation. Wenzhi and Selmic (2006) proposed a NN-based and feed-forward saturation compensation strategy for nonlinear systems with Brunovsky canonical form. In Wen et al. (2011), the Nussbaum function was employed to compensate for the nonlinear term caused by the input saturation. To handle the control issue for nonlinear systems with unknown saturation, an auxiliary system in He et al. (2016) and Peng et al. (2020) was proposed to tackle the actuator saturation. And in Zhao et al. (2018) a control strategy consisted of an ADP-based nominal control and a NN-based compensator was proposed. In Abu-Khalaf and Lewis (2005), the HJB equation was in the form of a non-quadratic function and NN least-squares method was proposed to obtain the solution.

In Peng et al. (2020), robot-environment interaction and actuator saturation are considered, while optimal control is not. However, for robot systems, it's worthwhile to investigate how to realize tracking control in an optimal manner. Therefore, based on our previous work, optimal tracking control issue for robot systems with environment interaction and actuator saturation will be studied in this paper. Inspired by Abu-Khalaf and Lewis (2005), Lyshevski (1998) and Jiang and Jiang (2012), a control scheme based on admittance control and ADP method is employed to improve the control performance of robot systems. The main contributions of this paper are summarized as follows

(i) To solve interaction problem, the unknown environment is regarded as a linear system and an admittance adaptation approach based on iterative linear quadratic regulator(LQR) is adopted to obtain the compliant behaviour of the robot.

(ii) To tackle the optimal tracking problem, an ADP-based controller is designed. The cost function is defined with non-quadratic form. The critic network with RBFNN is developed to derive an approximate solution to the minimum cost of the HJB equation, and then the corresponding optimal control is obtained.

The rest of this paper is arranged as follows. In Sect. 2, the robot systems with actuator saturation and the environment dynamics are described, and the control objective is provided. In Sect. 3, the control strategy based on admittance

adaptation and ADP-based optimal controller is proposed. In Sect. 4, simulation studies are performed on a 2-DOF planar manipulator. In Sect. 5, conclusion is drawn. The system stability is discussed and proved in Appendix.

2 Preliminaries and problem formulation

2.1 Robot dynamics

The dynamics of n-link robot manipulator subjected to actuator saturation is described as:

$$M(q)\ddot{q} + C(q, \dot{q})\dot{q} + G(q) = \mu \quad (1)$$

where $q \in \mathbb{R}^n$, $\dot{q} \in \mathbb{R}^n$, and $\ddot{q} \in \mathbb{R}^n$ denote the position, velocity and acceleration vectors in joint space of the robot, respectively. μ , λ and A denote the joint torque, admissible control set and constant saturation bound, respectively, and $\mu \in \lambda$, $\lambda = \{\mu \in \mathbb{R}^n : |\mu_i| \leq A\}$ is satisfied. For the sake of brevity, we use M , C and G to denote the known inertial matrix $M(q) \in \mathbb{R}^{n \times n}$, coriolis/centrifugal matrix $C(q, \dot{q}) \in \mathbb{R}^{n \times n}$ and gravity vector $G(q) \in \mathbb{R}^n$, respectively.

If we define the reference trajectory as $q_r \in \mathbb{R}^n$, the tracking error $q_e \in \mathbb{R}^n$ is given as $q_e = q - q_r$. Define the sliding motion surface as $\xi = \Lambda q_e + \dot{q}_e$, where $\Lambda \in \mathbb{R}^{n \times n}$ is a constant positive matrix, then we have

$$\begin{aligned} \dot{q} &= \xi - \Lambda q_e + \dot{q}_r \\ \ddot{q} &= \dot{\xi} - \Lambda \dot{q}_e + \ddot{q}_r \end{aligned} \quad (2)$$

According to (1) and (2), the error dynamics is derived as

$$\begin{aligned} \dot{\xi} &= -M^{-1}C(\xi - \Lambda q_e + \dot{q}_r) - M^{-1}G \\ &\quad - \ddot{q}_r + \Lambda \dot{q}_e + M^{-1}\mu \end{aligned} \quad (3)$$

Consequently, we can obtain the following system:

$$\dot{\xi} = f(\xi) + g(\xi)\mu \quad (4)$$

where $f : \mathbb{R}^n \rightarrow \mathbb{R}^n$ and $g : \mathbb{R}^n \rightarrow \mathbb{R}^{n \times n}$ are non-linear functions and described as

$$\begin{aligned} f(\xi) &= -M^{-1}C(\xi - \Lambda q_e + \dot{q}_r) - M^{-1}G - \ddot{q}_r + \Lambda \dot{q}_e \\ g(\xi) &= M^{-1} \end{aligned} \quad (5)$$

2.2 Environment dynamics

In this paper, we consider the unknown interaction environment which is regarded as a damping-stiffness model in Ge et al. (2014) given by

$$C_E \dot{x} + G_E x = -F \quad (6)$$

where C_E and G_E are unknown damping and stiffness of the environment, respectively. F represents the measured

interaction force by the force sensor and x is the end-effector position of the robot in Cartesian space.

We define x_d as the corresponding desired trajectory, and $U_d \in \mathbb{R}^{m \times m}$ is a known matrix, then $\dot{x}_d$ is expressed as follows

$$\dot{x}_d = U_d x_d \quad (7)$$

Consequently, define $\eta = [x, x_d]^T$, the dynamics of the environment and desired trajectory will be derived.

$$\begin{aligned} \dot{\eta} &= \begin{bmatrix} -C_E^{-1}G_E & \mathbf{0} \\ \mathbf{0} & U_d \end{bmatrix} \eta + \begin{bmatrix} -C_E^{-1} \\ \mathbf{0} \end{bmatrix} F \\ &= A_e \eta + B_e F \end{aligned} \quad (8)$$

Therefore, (8) can be regarded as a linear system, where F is the control input and η is the controlled state. $F = -K_e \eta$ is the corresponding optimal feedback control law and the object is to minimize the cost function given as

$$J_1 = \int_0^\infty (x_e^T Q_{E1} x_e + F^T R_E F) dt \quad (9)$$

From (9), we can see that the purpose of modifying trajectory x_d is to balance interaction force F and tracking error x_e defined as $x_e = x - x_d$, which can be realized by adjusting user-defined matrices Q_{E1} and R_E .

The robot dynamics with saturated actuator and unknown environment dynamics are described in this section. Then, an ADP enhanced admittance control scheme will be designed to ensure the complaint behaviour and optimal trajectory tracking with robot-environment interaction.

3 Control strategy

As shown in Fig.1, the designed control scheme inspired by Zhan et al. (2020) in this section consists of three parts, including an optimal trajectory modifier using admittance control to modify user-desired trajectory x_d to modified trajectory x_r , a closed loop inverse kinematics (CLIK) solver to transform x_r in Cartesian space into q_r in joint space and an optimal trajectory tracking controller based on ADP with the output torque μ acting on robot manipulator to ensure optimal tracking performance.

3.1 Trajectory modifier using admittance control

Formula (9) can be written as the following form by transformation, and the system counterpart is consistent with system (8).

Fig. 1 An illustration of the proposed control scheme

$$\Gamma = \int_0^\infty (\eta^T Q_E \eta + F^T R_E F) dt$$

$$Q_E = \begin{bmatrix} Q_{E1} & -Q_{E1} U_d \\ -U_d^T Q_{E1} & U_d^T Q_{E1} U_d \end{bmatrix} \quad (10)$$

It's noted that solving (10) can be regarded as the process similar to LQR problem. Then, the algebraic riccati equation (ARE) associated with (9) and (10) is given in (11). And in this subsection, an algorithm proposed by Jiang and Jiang (2012) is employed to solve the ARE and obtain the feedback gain K_e in (11).

$$P A_e + A_e^T P + Q_E - P B_e R_E^{-1} B_e^T P = 0$$

$$K_e = -R_E^{-1} B_e^T P \quad (11)$$

Now, we list the matrices with sampled signals as follows

$$\hat{p} = [p_{11}, 2p_{12}, \dots, 2p_{1n}, p_{22}, 2p_{23}, \dots, p_{nn}]^T$$

$$\bar{\eta} = [\eta_1^2, \eta_1 \eta_2, \dots, \eta_1 \eta_n, \eta_2^2, \eta_2 \eta_3, \dots, \eta_n^2]^T$$

$$d_{\bar{\eta}} = [\bar{\eta}(t_1) - \bar{\eta}(t_0), \bar{\eta}(t_2) - \bar{\eta}(t_1), \dots, \bar{\eta}(t_d) - \bar{\eta}(t_{d-1})]^T$$

$$I_{\eta}^n = \left[\int_{t_0}^{t_1} \eta \otimes \eta dt, \int_{t_1}^{t_2} \eta \otimes \eta dt, \dots, \int_{t_{d-1}}^{t_d} \eta \otimes \eta dt \right]^T$$

$$I_F^n = \left[\int_{t_0}^{t_1} \eta \otimes F dt, \int_{t_1}^{t_2} \eta \otimes F dt, \dots, \int_{t_{d-1}}^{t_d} \eta \otimes F dt \right]^T \quad (12)$$

where n, m, d denote the length of η, F and the sample times integer, respectively. p_{ij}, η_i represent entries of P and η , respectively. In addition, in (12), $\otimes$ represents the Kronecker product, and $p \in \mathbb{R}^{\frac{1}{2}n(n+1)}$, $\bar{\eta} \in \mathbb{R}^{\frac{1}{2}n(n-1)}$, $d_{\bar{\eta}} \in \mathbb{R}^{d \times \frac{1}{2}n(n-1)}$, $I_{\eta}^n \in \mathbb{R}^{d \times n^2}$, $I_F^n \in \mathbb{R}^{d \times nm}$.

Let $\| \cdot \|$ and $\text{vec}(\cdot)$ denote the 2-norm of $\cdot$ and the column vectorization of $\cdot$, respectively. Let k and $I_n \in \mathbb{R}^{n \times n}$ denote iteration index and an identity matrix, respectively. If the sampled data is large enough and the rank condition in (13) is satisfied, K_e can be solved by iteratively calculate (14) until $\|\hat{p}^{(k)} - \hat{p}^{(k-1)}\| < \varepsilon$, where ε is an acceptable range.

$$\text{rank} \left(\begin{bmatrix} I_{\eta}^n & I_F^n \end{bmatrix} \right) = \frac{n(n+1)}{2} + nm \quad (13)$$

$$Q_E^{(k)} = Q_E + K_e^{(k)T} R_E K_e^{(k)}$$

$$\Theta^{(k)} = \begin{bmatrix} d_{\bar{\eta}}, -2I_{\eta}^n (I_n \otimes K_e^{(k)T} R_E) - 2I_F^n (I_n \otimes R_E) \end{bmatrix}$$

$$\Xi^{(k)} = -I_{\eta}^n \text{vec}(Q_E^{(k)})$$

$$\begin{bmatrix} \hat{p}^{(k)} \\ \text{vec}(K_e^{(k+1)}) \end{bmatrix} = (\Theta^{(k)T} \Theta^{(k)})^{-1} \Theta^{(k)T} \Xi^{(k)} \quad (14)$$

When we obtain the optimal feedback gain K_e , the modified trajectory x_r which is to be tracked and equal to x in (15) can be calculated by (16), where K_{e1} and K_{e2} are compatible matrices of K_e .

$$F = -K_e \eta = -\begin{bmatrix} K_{e1} & K_{e2} \end{bmatrix} \begin{bmatrix} x \\ x_d \end{bmatrix} \quad (15)$$

$$x_r = -K_{e1}^{-1} F - K_{e1}^{-1} K_{e2} x_d \quad (16)$$

3.2 CLIK solver

We adopt the CLIK algorithm proposed by Siciliano 1990 to transform reference trajectory x_r in Cartesian space into q_r in joint space. Let $\kappa(\cdot)$ and K_f represent the forward kinematics and a positive user-defined matrix, respectively. Define $e := \kappa(q_r) - x_r$, $\dot{e} = -K_f e$, $\dot{x} = J_{co} \dot{q}$, $J_{co} = \partial \kappa(q) / \partial q$, then

$$\dot{q}_r = J_{co}^{\dagger} (\dot{x}_r - K_f (\kappa(q_r) - x_r)) \quad (17)$$

Integrating both sides of the above equation, q_r can be obtained as follows

$$q_r = \int_0^t (J_{co}^{\dagger} \dot{x}_r - J_{co}^{\dagger} K_f (\kappa(q_r) - x_r)) dt \quad (18)$$

where $q(0) = \kappa^{-1}(x_r(0))$, $J_{co}^{\dagger} = J_{co}^T (J_{co} J_{co}^T + \sigma I_n)^{-1}$, and $\sigma \in \mathbb{R}$. Note that σ is used to prevent the singularity problem and it is also required to be small enough to promote the accuracy of the solution.

3.3 Optimal control using ADP

The objective of this section is to find the stabilizing control input μ of the robot system (4) which could minimize the defined cost function. According to optimal theory, the optimal feedback control of system (4) can be obtained by solving HJB equation in ADP framework. The structure diagram of the ADP-based tracking controller is given in Fig. 2.

We assume that system (4) is controllable, the nonlinear functions $f(\xi)$ and $g(\xi)$ are Lipschitz continuous and differentiable in $\mathbb{R}^{2n}$. In order to deal with actuator saturation of the robot system, inspired by Abu-Khalaf and Lewis (2005) and Lyshevski (1998), we define the cost function as follows

$$J(\xi(t)) = \int_t^\infty [\Phi(\xi(s)) + U(\xi(s), \mu(\xi(s)))] ds \quad (19)$$

where

$$\Phi(\xi(s)) = \xi(s)^T Q \xi(s) \quad (20)$$

$$U(\xi(s), \mu(\xi(s))) = 2A \int_0^\mu (\Psi^{-1}(v/A))^T R dv \quad (21)$$

It is noted that $Q \in \mathbb{R}^{n \times n}$ in (20) is symmetric positive definite. In (21), $\Psi^{-1}(v/A) = [\psi^{-1}(v_1/A), \psi^{-1}(v_2/A), \dots, \psi^{-1}(v_n/A)]^T$, $\Psi \in \mathbb{R}^n$. $\psi(\cdot)$ is a strictly monotonic odd function and its first derivative is bounded by a constant B . Meanwhile, R is also a symmetric and positive definite matrix. Therefore, $U(\xi(s), \mu(\xi(s)))$ is also positive definite. Without loss of generality, we select $\psi(\cdot) = \tanh(\cdot)$ and $R = rI_n$ with r as a positive constant and I_n as the identity matrix of n -dimension.

If $J(\xi(t))$ defined in (19) is continuously differentiable, by taking the time derivative of (19), we can get the following nonlinear Lyapunov equation with $J(0) = 0$, which is an infinitesimal form of (19).

$$0 = \Phi(\xi) + 2Ar \int_0^\mu (\tanh^{-1}(v/A))^T dv + (\nabla J(\xi))^T (f(\xi) + g(\xi)\mu(\xi)) \quad (22)$$

where $J(\xi)$ denotes $J(\xi(t))$ and $\nabla * \triangleq \frac{\partial *}{\partial \xi}$ denotes the partial derivative of $*$ for convenience.

Therefore, the Hamiltonian function and optimal cost function are described as

$$H(\xi, \mu(\xi), \nabla J(\xi)) = \Phi(\xi) + 2Ar \int_0^\mu (\tanh^{-1}(v/A))^T dv + (\nabla J(\xi))^T (f(\xi) + g(\xi)\mu(\xi)) \quad (23)$$

$$J(\xi)^* = \min_{\mu \in \lambda} \int_t^\infty [\Phi(\xi(s)) + U(\xi(s), \mu(\xi(s)))] ds \quad (24)$$

We can derive HJB equation as below

$$0 = \min_{\mu \in \lambda} H(\xi, \mu(\xi), \nabla J^*(\xi)) \quad (25)$$

Suppose that the minimum value on the right side of formula (25) exists and also is unique, then from $\frac{\partial H(\xi, \mu(\xi), \nabla J^*(\xi))}{\partial \mu} = 0$, we can obtain the optimal control $\mu^*(\xi)$ as

$$\mu^*(\xi) = -A \tanh \left(\frac{1}{2A} r^{-1} g^T(\xi) \nabla J^*(\xi) \right) \quad (26)$$

Substituting (26) into (22), another HJB equation form related to $\nabla J^*(\xi)$ will be derived as

$$H(\xi, \mu^*(\xi), \nabla J^*(\xi)) = 0 \quad (27)$$

Then, from (26) and (27), the HJB equation for the robot system with actuator saturation becomes

$$\begin{aligned} H(\xi, \mu^*(\xi), \nabla J^*(\xi)) &= (\nabla J^*(\xi))^T f(\xi) \\ &\quad - 2A^2 r D^T(\xi) \tanh(D(\xi)) \\ &\quad + \Phi(\xi) + 2Ar \int_0^{-A \tanh(D(\xi))} \tanh^{-1}(v/A) dv = 0 \end{aligned} \quad (28)$$

where $D(\xi) = \frac{1}{2A} r^{-1} g^T(\xi) \nabla J^*(\xi)$. Applying the integral formula of inverse hyperbolic function, we have

Fig. 2 Structure diagram of the ADP-based tracking controller

$$\begin{aligned}
& 2Ar \int_0^{-A \tanh(D(\xi))} \tanh^T(v/A) dv \\
&= 2Ar \sum_{i=1}^n \int_0^{-A \tanh(D_i(\xi))} \tanh^T(v_i/A) dv_i \\
&= 2A^2 r D^T(\xi) \tanh(D(\xi)) \\
&\quad + A^2 r \sum_{i=1}^n \ln[1 - \tanh^2(D_i(\xi))]
\end{aligned} \quad (29)$$

where $D(\xi) = (D_1(\xi), \dots, D_n(\xi))^T$ with $D_i(\xi) \in \mathbb{R}, i = 1, \dots, n$. Substituting (29) into (28), (28) can be rewritten as follows

$$\begin{aligned}
H(\xi, \mu^*(\xi), \nabla J^*(\xi)) &= (\nabla J^*(\xi))^T f(\xi) \\
&+ \Phi(\xi) + A^2 r \sum_{i=1}^n \ln[1 - \tanh^2(D_i(\xi))] = 0
\end{aligned} \quad (30)$$

However, (30) is a nonlinear partial differential equation with regard to $J^*(\xi)$ and it's very difficult to obtain $J^*(\xi)$ from it, even impossible.

Suppose $J^*(\xi)$ is continuously differentiable, it can be constructed by RBFNN and described by

$$J^*(\xi) = w^T S(\xi) + \varepsilon(\xi) \quad (31)$$

where $w \in \mathbb{R}^l$ and $S : \mathbb{R}^{2n} \rightarrow \mathbb{R}^l$ represent the ideal constant weight and the activation function, respectively. l and $\varepsilon(\xi)$ denote the node number of the hidden layer and the unknown approximation error of the critic NN, respectively. Consequently, we can obtain the derivation of (31) refer to ξ as follows.

$$\nabla J^*(\xi) = (\nabla S(\xi))^T w + \nabla \varepsilon(\xi) \quad (32)$$

From (26) and (32) and using Taylor series expansion, we have μ^* shown as

$$\mu^*(\xi) = -A \tanh\left(\frac{1}{2A} r^{-1} g^T(\xi) (\nabla S(\xi))^T w\right) + \varepsilon_{\mu^*} \quad (33)$$

$$\varepsilon_{\mu^*} = -\frac{1}{2} (1 - \tanh^2(\iota)) g^T(\xi) \nabla \varepsilon(\xi) \quad (34)$$

where $\mathbf{1} = (1, \dots, 1)^T \in \mathbb{R}^n$ and $\iota \in \mathbb{R}^n$ is selected between $\frac{1}{2A} r^{-1} g^T(\xi) (\nabla S(\xi))^T w$ and $\frac{1}{2A} r^{-1} g^T(\xi) ((\nabla S(\xi))^T w + \nabla \varepsilon(\xi))$. Then, by substituting (32) into (30), (30) will be written as

$$\begin{aligned}
H^*(\xi, \mu^*(\xi), \nabla J^*(\xi)) &= w^T (\nabla S(\xi)) f(\xi) + \Phi(\xi) \\
&\quad + A^2 r \sum_{i=1}^n \ln[1 - \tanh^2(B_{1i}(\xi))] \\
&\quad + \varepsilon_{HJB} = 0
\end{aligned} \quad (35)$$

$$B_1(\xi) = \frac{1}{2A} r^{-1} g^T(\xi) (\nabla S(\xi))^T w \quad (36)$$

where $B_1(\xi) = (B_{11}(\xi), \dots, B_{1n}(\xi))^T$, $B_{1i} \in \mathbb{R}$ and ε_{HJB} is the HJB approximation error.

Actually, the ideal w and $J^*(\xi)$ in (31) are not available, so we can derive the estimated weight and optimal cost function which are represented by $\hat{w}$ and $\hat{J}(\xi)$ respectively by the constructed critic NN described as

$$\hat{J}(\xi) = \hat{w}^T S(\xi) \quad (37)$$

Then, the partial derivative of $\hat{J}(\xi)$ refers to ξ and the approximate optimal control $\hat{\mu}(\xi)$ can be obtained as follows

$$\nabla \hat{J}(\xi) = (\nabla S(\xi))^T \hat{w} \quad (38)$$

$$\hat{\mu}(\xi) = -A \tanh\left(\frac{1}{2A} r^{-1} g^T(\xi) (\nabla S(\xi))^T \hat{w}\right) \quad (39)$$

According to (23), (38) and (39), we can obtain the approximate Hamiltonian function $\hat{H}(\xi, \hat{\mu}_n(\xi), \nabla \hat{J}(\xi))$ shown as

$$\begin{aligned}
\hat{H}(\xi, \hat{\mu}(\xi), \nabla \hat{J}(\xi)) &= \hat{w}^T (\nabla S(\xi)) f(\xi) + \Phi(\xi) \\
&\quad + A^2 r \sum_{i=1}^n \ln[1 - \tanh^2(B_{2i}(\xi))]
\end{aligned} \quad (40)$$

$$B_2(\xi) = \frac{1}{2A} r^{-1} g^T(\xi) (\nabla S(\xi))^T \hat{w} \quad (41)$$

where $B_2(\xi) = (B_{21}(\xi), \dots, B_{2n}(\xi))^T$, $B_{2i}(\xi) \in \mathbb{R}$. Now we define the approximate neural network weight error as $\tilde{w} = w - \hat{w}$, and the error between $\hat{H}$ and H^* as E_H , then, we have

$$\begin{aligned}
E_H &= \hat{H}(\xi, \hat{\mu}(\xi), \nabla \hat{J}(\xi)) - H^*(\xi, \mu^*(\xi), \nabla J^*(\xi)) \\
&= \hat{H}(\xi, \hat{\mu}(\xi), \nabla \hat{J}(\xi)) \\
&= -\tilde{w}^T \nabla S(\xi) f(\xi) \\
&\quad + A^2 r \sum_{i=1}^n [Y(B_{2i}(\xi)) - Y(B_{1i}(\xi))] - \varepsilon_{HJB}
\end{aligned} \quad (42)$$

where $Y(B_{\ell i}(\xi)) = \ln[1 - \tanh^2(B_{\ell i}(\xi))]$, $\ell = 1, 2$ and $i = 1, \dots, n$. Note that $Y(B_{\ell i}(\xi))$ can be expressed as

$$\begin{aligned}
Y(B_{\ell i}(\xi)) &= \ln[1 - \tanh^2(B_{\ell i}(\xi))] \\
&= \begin{cases} \ln 4 - 2B_{\ell i}(\xi) - 2 \ln(1 + \exp(-2B_{\ell i}(\xi))), & B_{\ell i}(\xi) > 0 \\ \ln 4 + 2B_{\ell i}(\xi) - 2 \ln(1 + \exp(2B_{\ell i}(\xi))), & B_{\ell i}(\xi) < 0 \end{cases}
\end{aligned} \quad (43)$$

For convenience, it can be written as follows

$$\begin{aligned}
Y(B_{\ell i}(\xi)) &= \ln 4 - 2B_{\ell i}(\xi) \operatorname{sgn}(B_{\ell i}(\xi)) - 2 \\
&\quad \ln[1 + \exp(-2B_{\ell i}(\xi) \operatorname{sgn}(B_{\ell i}(\xi)))]
\end{aligned} \quad (44)$$

where $\text{sgn}(B_{\ell i}(\xi))$ is the sign function.

To train the critic NN, inspired by Liu et al. (2017) and Yang et al. (2013), a suitable weight updating law $\hat{w}$ is designed, which can minimize the objective function $E_c = \frac{1}{2}E_H^2$ and also ensure that $\hat{w}$ converges to w .

$$\begin{aligned} \dot{\hat{w}} = & -\alpha_H \bar{\phi}(\Phi(\xi)) \\ & + \hat{w}^T \nabla S(\xi) f(x) + A^2 r \sum_{i=1}^n \ln[1 - \tanh^2(B_{2i}(\xi))] \\ & + \frac{\alpha_H}{2} h \nabla S(\xi) g(\xi) [I_n - Z(B_2(\xi))] g^T(\xi) \nabla V_s(\xi) \\ & + \alpha_H (A \nabla S(\xi) g(\xi) [\tanh(B_2(\xi)) - \text{sgn}(B_2(\xi))] \frac{\phi^T}{m_s} \hat{w} \\ & - (F_2 \hat{w} - F_1 \phi^T \hat{w})) \end{aligned} \quad (45)$$

where $\bar{\phi} = \phi/m_s^2$, $m_s = 1 + \phi^T \phi$, $\phi = \nabla S(\xi) f(\xi) - A \nabla S(\xi) g(\xi) \tanh(B_2(\xi))$, $\phi = \phi/m_s$, $\alpha_H > 0$ is a designed parameter, $Z(B_2(\xi)) = \text{diag}[\tanh^2(B_{21}(\xi)), \dots, \tanh^2(B_{2n}(\xi))]$ and F_1 and F_2 are tuning parameters with suitable dimensions. In (45), h is described as follows:

$$h = \begin{cases} 0, & \text{if } (\nabla V_s(\xi))^T (f(\xi) - A g(\xi) \tanh(B_2(\xi))) < 0 \\ 1, & \text{else} \end{cases} \quad (46)$$

where $V_s(\xi)$ is chosen as a Lyapunov function candidate which is continuously differentiable. Suppose that one positive definite matrix N exists, we have the following formula satisfied.

$$\begin{aligned} \dot{V}_s(\xi) &= (\nabla V_s(\xi))^T (f(\xi) + g(\xi) \mu^*) \\ &= -(\nabla V_s(\xi))^T N \nabla V_s(\xi) < 0 \end{aligned} \quad (47)$$

Here, $V_s(\xi)$ is a polynomial with regard to the state variable ξ , which can be appropriately selected, such as $V_s(\xi) = \frac{1}{2} \xi^T k_\xi \xi$.

Remark 1 The $\dot{\hat{w}}$ in (45) composes of two parts: the first term is based on the standard gradient descent algorithm and the others are introduced to ensure the stability of the robot system during the critic NN learning process. Note that in (46), if $(\nabla V_s(\xi))^T (f(\xi) - A g(\xi) \tanh(B_2(\xi))) \geq 0$, the system is unstable, then $h = 1$ and the second term in (46) will be activated, which improves the learning process. Therefore, the initial stabilizing control requirement will be released.

Remark 2 From (40) and (45), we can see that if $x = 0$ and $f(x) = 0$, then $\hat{H}(\xi, \hat{\mu}(\xi), \nabla \hat{J}(\xi)) = 0$. If $F_2 = F_1 \phi^T$, then we have $\dot{\hat{w}} = 0$ and the critic NN will not be updated and the optimal control may not be obtained. Consequently, the persistence excitation is required.

3.4 Stability analysis

We will discuss the system stability of the robot and give detailed proof that the estimated weight error $\tilde{w}$ and the system state ξ are ultimately uniformly bounded.

Now we give the necessary assumption as follows:

Assumption There exist known positive constants w_m , ε_M , ε_N such that $\|w\| \leq w_m$, $\|\varepsilon\| \leq \varepsilon_M$, $\|\varepsilon_{u^*}\| \leq \varepsilon_N$, respectively. Item $g(\xi)$ in (4) is bounded over a compact set Ω , i.e., there exist positive constants g_m and g_M such that $g_m \leq \|g(\xi)\| \leq g_M$.

Theorem Considering the robot system (1) referring to actuator saturation, the corresponding HJB equation (30) and Assumption, if the control law is designed as (39) and the critic NN weight update according to (45), it can be concluded that the critic NN weight approximation error $\tilde{w}$ and the state ξ are guaranteed to be ultimately uniformly bounded(UUB).

Proof see the Appendix. $\square$

4 Simulation study

4.1 Simulation settings

A two-link manipulator, constructed by the robotics toolbox in Corke (2017) and shown in Fig. 3, is employed to verify the proposed control strategy, whose dynamic parameters are given in Table 1. The simulation runs on the Matlab 2018a software with an ode3 solver and the fixed time step is 0.01s. The robot manipulator is required to track a reference trajectory and interact simultaneously with a virtual environment govern by

$$F = \begin{cases} -C_E \dot{x} - G_E(x - x_0) & x \leq x_0 \\ 0 & x > x_0 \end{cases} \quad (48)$$

where $C_E = 0.1$, $G_E = 1.0$, x_0 denotes the contour of an object and F denotes the reactive force due to the penetration into the object. For simplicity and generality, only the trajectory along x-axis is modified and disturbed by the external interaction forces.

Table 1 Parameters of the robot manipulator

Item	Value of link 1	Value of link 2
Length of the link	0.50 m	0.50 m
Mass of the link	5.00 kg	5.00 kg
Initial joint position	0.08211 rad	1.89 7rad

Fig. 3 An illustration of the simulation settings

Parameters of the proposed control scheme are selected as follows: for the “Optimal Trajectory Modifier” block in Fig. 1, in (10), $Q_{E1} = 1.0$, $R_E = 1.0$, the reference trajectory is $x_d = [0.3e^{-0.5t}, 0.5]^T m$, where $U_d = 0.3$; the feedback gain of the inverse kinematics in (18), $K_f = 30$, $\sigma = 1e - 6$. An RBFNN is selected to approximate the cost function in (31), where $S(\xi) = \exp((\xi - c)^T(\xi - c)/\sigma_N^2)$ with $\hat{w} \in \mathbb{R}^{49}$, $S(\xi) \in \mathbb{R}^{49}$. For the controller in (39), $A = 6N \cdot m$, the centers and variance of the RBFNN are $c \in [-1.5, -0.5, -0.1, 0, 0.1, 0.5, 1.5] \times [-1.5, -0.5, -0.1, 0, 0.1, 0.5, 1.5]$, $\sigma_N = 0.6$, $\hat{w}(0) = \mathbf{0}$. For the updated law in (45), $V_s = 2\xi^T \xi$, $\alpha_H = 30$, $Q = 200$, $R = 0.006$, $F_1 = 1e - 6$, $F_2 = 1e - 8$.

4.2 Results analysis

The control performance is shown in Fig. 4, from which we can see that at the beginning of the control, there exists a large transient error since the weights of the RBFNN have not converged. However, before the trajectory starts to be modified at $t = 4.2$ s, the tracking error has reduced to an acceptable range. Subsequently, the actual trajectory gradually converges to the desired trajectory. Fig. 5 gives the control signals during the control process. In this figure, we can clearly see that the control input stays within the limits of the actuator and weights of the RBFNN eventually converge to constant values. These observations demonstrate the effectiveness of the ADP-based controller under the saturation effect.

To show the effectiveness of the optimal admittance adaptation control, the control performance under two different feedback gain K_e that affects the trajectory modification in (16) is compared, wherein K_e^{opt} is obtained by assuming that the dynamic parameters of the environment in (6) are exactly known, while K_e^{pro} is calculated by the algorithm presented in (14). Note that unlike the virtual environment used in (48), environmental dynamics in (6) adopted for theoretical design does not take the contour of the environment x_0 into consideration. Thus, K_e^{opt} is sub-optimal. The results are shown in Fig. 6. We can notice that both the tracking error and value of the cost function in (9) under K_e^{pro} are smaller than those under K_e^{opt} , which shows the superiority of the proposed method when dynamics of the environment are unknown.

Fig. 4 Tracking performance of the proposed control scheme. Up: reference and modified trajectory during the interaction. Down: tracking errors

5 Conclusion

In this paper, the optimal tracking control issue for robot systems with environment interaction and actuator saturation is addressed. An ADP-based controller enhanced admittance adaptation control scheme is developed. The unknown environment is considered as a linear system and admittance adaptation control ensures the compliant behaviour of the robot. In ADP-based controller, to guarantee the optimal tracking performance, RBFNN is used to approach to the minimum cost function and make an optimal control of the HJB equation. The system stability is analysed and the simulation studies are performed to demonstrate the effectiveness of this control scheme.

Other input constraints such as dead zones and hysteresis, and dynamic uncertainties are also very common in actual robotic systems. These constraints will not only reduce the system performance, but also affect the system stability. Consequently, under ADP framework, the optimal control with other constraints and dynamic uncertainties will be considered in our future work.

Fig. 5 Control signals of the proposed control scheme. Up: control input of the controller. Down: weights of the RBFNN

Appendix

Stability analysis

This appendix demonstrates the stability of the ADP-based controller proposed in this paper for robot systems with actuator saturation. The Lyapunov candidate is selected as follows (Liu et al. 2017)

$$V(\xi) = V_s(\xi) + \frac{1}{2} \tilde{W}^T \alpha_H^{-1} \tilde{W} \quad (49)$$

From (49) and (39), the derivative of $V(\xi)$ can be derived as

$$\dot{V}(\xi) = (\nabla V_s)^T(\xi)(f(\xi) - Ag(\xi)\tanh(B_2(\xi))) + \dot{\tilde{W}}^T \alpha_H^{-1} \tilde{W}. \quad (50)$$

Next we will calculate the last term in (50). Note that

$$\begin{aligned} \sum_{i=1}^n Y(B_{\ell i}(\xi)) \\ = n \ln 4 - 2B_{\ell}^T(\xi) \text{sgn}(B_{\ell}(\xi)) \\ - 2 \sum_{i=1}^n \ln [1 + \exp(-2B_{\ell i}(\xi) \text{sgn}(B_{\ell i}(\xi)))] \end{aligned} \quad (51)$$

Fig. 6 A comparison for the modified tracking performance under the optimal feedback gain K_e^{opt} in (11) and feedback gain obtained from the proposed control scheme K_e^{pro} . Up: time series of the cost function in (9); Down: tracing performance

From (40) and (51), we have

$$\begin{aligned} \hat{H}(\xi, \hat{\mu}(\xi), \nabla \hat{J}(\xi)) \\ = 2A^T r [B_1^T(\xi) \text{sgn} B_1(\xi) - B_2^T(\xi) \text{sgn} B_2(\xi)] \\ - \tilde{W}^T \nabla S(\xi) f(\xi) \\ + A^2 r \Delta_B - \epsilon_{HJB} \\ = A[w^T \nabla S(\xi) g(\xi) \text{sgn} B_1(\xi) \\ - \hat{W}^T \nabla S(\xi) g(\xi) \text{sgn} B_2(\xi)] \\ - \tilde{W} \nabla S(\xi) f(\xi) + A^2 r \Delta_B - \epsilon_{HJB} \\ = -\tilde{W} \nabla S(\xi) f(\xi) \\ + A \tilde{W}^T \nabla S(\xi) g(\xi) \text{sgn} B_2(\xi) + D_1(\xi) \end{aligned} \quad (52)$$

where

$$\Delta_B = 2 \sum_{i=0}^n \ln \frac{1 + \exp[-2B_{1i}(\xi) \text{sgn}(B_{1i}(\xi))]}{1 + \exp[-2B_{2i}(\xi) \text{sgn}(B_{2i}(\xi))]} \quad (53)$$

$$D_1(\xi) = A w^T \nabla S(\xi) g(\xi) [\operatorname{sgn}(B_1(\xi)) - \operatorname{sgn}(B_2(\xi))] + A^2 r \Delta_B - \varepsilon_{HJB} \quad (54)$$

From ϕ given in (45), we have $\phi = \nabla S(\xi) f(\xi) - A \nabla S(\xi) g(\xi) \tanh(B_2(\xi))$. Then, (52) becomes

$$\hat{H}(\xi, \hat{\mu}(\xi), \nabla \hat{J}(\xi)) = -\tilde{w}^T \phi + A \tilde{w}^T \nabla S(\xi) g(\xi) T(\xi) + D_1(\xi) \quad (55)$$

where $T(\xi) = \operatorname{sgn}(B_2(\xi)) - \tanh(B_2(\xi))$.

Based on (40), (45) and (55), we have

$$\begin{aligned} \dot{\hat{w}} &= \alpha_H \frac{\varphi}{m_s} [-\tilde{w}^T \phi + A \tilde{w}^T \nabla S(\xi) g(\xi) T(\xi) + D_1(\xi)] \\ &\quad - \frac{\alpha_H}{2} h \nabla S(\xi) g(\xi) [I_n - Z(B_2(\xi))] g^T(\xi) \nabla V_s(\xi) \\ &\quad + \alpha_H [A \nabla S(\xi) g(\xi) T(\xi) \frac{\varphi^T}{m_s} \hat{w} + (F_2 \hat{w} - F_1 \varphi^T \hat{w})] \end{aligned} \quad (56)$$

Consequently, the last term in (50) can be expressed as follows

$$\begin{aligned} \dot{\hat{w}}^T \alpha_H^{-1} \tilde{w} &= [-\tilde{w}^T \phi + A \tilde{w}^T \nabla S(\xi) g(\xi) T(\xi) + D_1(\xi)] \frac{\varphi^T}{m_s} \tilde{w} \\ &\quad - \frac{1}{2} h (\nabla V_s(\xi))^T g(\xi) [I_m - Z(B_2(\xi))] g^T(\xi) (\nabla S(\xi))^T \tilde{w} \\ &\quad + A \tilde{w}^T \nabla S(\xi) g(\xi) T(\xi) \frac{\varphi^T}{m_s} \hat{w} \\ &\quad + \tilde{w}^T (F_2 \hat{w} - F_1 \varphi^T \hat{w}) \\ &= -\tilde{w}^T \varphi \varphi^T \tilde{w} + \bar{D}_1(\xi) \varphi^T \tilde{w} + \tilde{w}^T \bar{D}_2(\xi) \\ &\quad - \frac{1}{2} h (\nabla V_s(\xi))^T g(\xi) [I_n - Z(B_2(\xi))] g^T(\xi) (\nabla S(\xi))^T \tilde{w} \\ &\quad + \tilde{w}^T (F_2 \hat{w} - F_1 \varphi^T \hat{w}) \end{aligned} \quad (57)$$

where $\bar{D}_1(\xi) = \frac{D_1(\xi)}{m_s}$, $\bar{D}_2(\xi) = A \nabla S(\xi) g(\xi) T(\xi) \frac{\varphi^T}{m_s} w$.

Applying $\hat{w} = w - \tilde{w}$, we have

$$\tilde{w}^T (F_2 \hat{w} - F_1 \varphi^T \hat{w}) = \tilde{w}^T F_2 w - \tilde{w}^T F_2 \tilde{w} - \tilde{w}^T F_1 \varphi^T w + \tilde{w}^T F_1 \varphi^T \tilde{w} \quad (58)$$

Substituting (58) into (57) and defining $\beta^T = [\tilde{w}^T \varphi, \tilde{w}^T]$, (57) can be written as

$$\begin{aligned} \dot{\hat{w}}^T \alpha_H^{-1} \tilde{w} &= -\beta^T W_1 \beta + \beta^T W_2 \\ &\quad - \frac{1}{2} h (\nabla V_s(\xi))^T g(\xi) [I_n - Z(B_2(\xi))] g^T(\xi) (\nabla S(\xi))^T \tilde{w} \end{aligned} \quad (59)$$

where $W_1 = \begin{bmatrix} I & -\frac{1}{2} F_1^T \\ -\frac{1}{2} F_1 & F_2 \end{bmatrix}$, $W_2 = \begin{bmatrix} \bar{D}_1(\xi) \\ \bar{D}_2(\xi) + F_2 w - F_1 \varphi^T w \end{bmatrix}$.

From (59) and (50), if we choose appropriate F_1 and F_2 to make W_1 positive definite, the following result will be derived.

$$\begin{aligned} \dot{V}(\xi) &\leq (\nabla V_s(\xi))^T (f(\xi) - Ag(\xi) \tanh(B_2(\xi))) - \rho_{\min}(W_1) \|\beta\|^2 \\ &\quad + b_m \|\beta\| - \frac{1}{2} h (\nabla V_s(\xi))^T g(\xi) [I_m - Z(B_2(\xi))] g^T(\xi) (\nabla S(\xi))^T \tilde{w} \end{aligned} \quad (60)$$

where $\rho_{\min}(\ast)$ denotes the minimum eigenvalue of matrix $\ast$ and b_m is the upper bound of $\|W_2\|$.

Case One: $h = 0$, that is $(\nabla V_s(\xi))^T (f(\xi) - Ag(\xi) \tanh(B_2(\xi))) < 0$. Since $\|\xi\| > 0$, then there exist a constant a_s such that $0 < a_s \leq \|\xi\|$ implies $\nabla V_s(\xi)^T \xi \leq -a_s \|\nabla V_s(\xi)\|$. Consequently, we can obtain

$$\begin{aligned} \dot{V}(\xi) &\leq -a_s \|\nabla V_s(\xi)\| \\ &\quad - \rho_{\min}(W_1) (\|\beta\| - \frac{b_m}{2\rho_{\min}(W_1)})^2 \\ &\quad + \frac{b_m^2}{4\rho_{\min}(W_1)} \end{aligned} \quad (61)$$

From (61), we can see that if one of the following conditions is satisfied, then $\dot{V}(\xi) < 0$ will be obtained.

$$\|\nabla V_s(\xi)\| > \frac{b_m^2}{4a_s \rho_{\min}(W_1)}, \quad \text{or} \quad \|\beta\| > \frac{b_m}{\rho_{\min}(W_1)} \quad (62)$$

Note that $\frac{a}{(1+a)^2} \leq \frac{1}{4}$, $\forall a$, while $\|\varphi\|^2 = \frac{\phi^T \phi}{(1+\phi^T \phi)^2}$, then we have $\|\phi\| \leq \frac{1}{2}$. From the definition of β , we can obtain $\|\beta\| \leq \sqrt{1 + \|\varphi\|^2} \|\tilde{w}\| \leq \frac{\sqrt{5}}{2} \|\tilde{w}\|$. Consequently, From (62), we have

$$\|\tilde{w}\| > \frac{2b_m}{\sqrt{5}\rho_{\min}(W_1)} \quad (63)$$

Case Two: $h = 1$, that is $(\nabla V_s(\xi))^T (f(\xi) - Ag(\xi) \tanh(B_2(\xi))) \geq 0$, then (60) becomes

$$\begin{aligned}\dot{V}(\xi) &\leq (\nabla V_s(\xi))^T f(\xi) - A(\nabla V_s(\xi))^T g(\xi)(\tanh(B_2(x)) \\ &\quad + \frac{1}{2A}[I_m - Z(B_2(\xi))]g^T(\xi)(\nabla S(\xi))^T \tilde{w}) \\ &\quad - \rho_{\min}(W_1)(\|\beta\| - \frac{b_m}{2\rho_{\min}(W_1)})^2 + \frac{b_m^2}{4\rho_{\min}(W_1)}\end{aligned}\quad (64)$$

Using the Taylor series expansion, we have

$$\begin{aligned}\tanh(B_1(\xi)) - \tanh(B_2(\xi)) \\ &= \tanh(B_2(\xi))(B_1(\xi) - B_2(\xi)) + O((B_1(\xi) - B_2(\xi))^2) \\ &= \frac{1}{2A}[I_n - Z(B_2(\xi))]g^T(\xi)(\nabla S(\xi))^T \tilde{w} \\ &\quad + O((B_1(\xi) - B_2(\xi))^2)\end{aligned}\quad (65)$$

Then, we can get

$$\begin{aligned}\tanh(B_2(\xi)) + \frac{1}{2A}[I_n - Z(B_2(\xi))]g^T(\xi)(\nabla S(\xi))^T \tilde{w} \\ = \tanh(B_1(\xi)) - O((B_1(\xi) - B_2(\xi))^2)\end{aligned}\quad (66)$$

Substituting (66) into (64), we have

$$\begin{aligned}\dot{V}(\xi) &\leq (\nabla V_s(\xi))^T (f(\xi) + g(\xi)\mu^*) - V_s^T g(\xi)\varepsilon_{\mu^*} + \\ &\quad A(\nabla V_s(\xi))^T g(\xi)O((B_1(\xi) - B_2(\xi))^2) - \\ &\quad \rho_{\min}(W_1)(\|\beta\| - \frac{b_m}{2\rho_{\min}(W_1)})^2 + \frac{b_m^2}{4\rho_{\min}(W_1)}\end{aligned}\quad (67)$$

According to the Assumption, (67) can be rewritten as

$$\begin{aligned}\dot{V}(\xi) &\leq -\rho_{\min}(N)(\|V_s\| \\ &\quad - \frac{\lambda}{2\rho_{\min}(N)})^2 - \rho_{\min}(W_1)(\|\beta\| \\ &\quad - \frac{b_m}{2\rho_{\min}(W_1)}) + \varepsilon_0\end{aligned}\quad (68)$$

where $\lambda = g_M(\varepsilon_N + A\varepsilon_m)$, ε_m is the upper bound of $O((B_1(\xi) - B_2(\xi))^2)$, and ε_0 is shown as follows

$$\varepsilon_0 = \frac{\lambda^2}{4\rho_{\min}(N)} + \frac{b_m^2}{4\rho_{\min}(W_1)}\quad (69)$$

From (68), we can see that if one of the following conditions is satisfied, then $\dot{V}(\xi) < 0$ will be obtained.

$$\begin{aligned}\|V_s(\xi)\| &> \frac{\lambda}{2\rho_{\min}(N)} \\ &\quad + \sqrt{\frac{\varepsilon_0}{\rho_{\min}(N)}} \\ \text{or } \|\beta\| &> \frac{b_m}{2\rho_{\min}(W_1)} \\ &\quad + \sqrt{\frac{\varepsilon_0}{\rho_{\min}(W_1)}}\end{aligned}\quad (70)$$

From $\|\beta\| \leq \frac{\sqrt{5}}{2}\|\tilde{w}\|$ and (70), we have

$$\|\tilde{w}\| > \frac{b_m}{\sqrt{5}\rho_{\min}(W_1)} + 2\sqrt{\frac{\varepsilon_0}{5\rho_{\min}(W_1)}}\quad (71)$$

According to the Lyapunov theorem and combining Case One and Case Two, it's concluded that the NN weight approximation error $\tilde{w}$ and function $V_s(\xi)$ are UUB. Since $V_s(\xi)$ is a selected polynomial with regard to ξ , we can concluded that the state ξ is also UUB. This completes the stability analysis.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

References

- Abu-Khalaf, M., Lewis, F.L.: Nearly optimal control laws for nonlinear systems with saturating actuators using a neural network hjb approach. *Automatica* **41**(5), 779–791 (2005)
- Bellman, R.: *Dynamic programming*. Princeton University Press, Princeton (1957)
- Braun, D., Petit, F., Huber, F., Haddadin, S., van der Smagt, P., Albu-Schaffer, A., Vijayakumar, S.: Optimal torque and stiffness control in compliantly actuated robots. pp. 2801–2808 (2012)
- Cervantes, I., Alvarez-Ramirez, J.: On the pid tracking control of robot manipulators. *Syst. Control Lett.* **42**(1), 37–46 (2001)
- Cohen, M., Flash, T.: Learning impedance parameters for robot control using an associative search network. *IEEE Trans Robot Autom* **7**, 382–390 (1991)
- Corke, P.: *Robotics, vision and control: fundamental algorithms in MATLAB® second, completely revised*, vol. 118. Springer, New York (2017)
- Cui, X., Zhang, H., Luo, Y., Jiang, H.: Adaptive dynamic programming for tracking design of uncertain nonlinear systems with disturbances and input constraints. *Int. J. Adapt. Control Signal Process.* **31**(11), 1567–1583 (2017)
- Ge, S.S., Li, Y., Wang, C.: Impedance adaptation for optimal robot-environment interaction. *Int. J. Control* **87**(2), 249–263 (2014)
- He, W., Dong, Y., Sun, C.: Adaptive neural impedance control of a robotic manipulator with input saturation. *IEEE Trans. Syst. Man Cybern. Syst.* **46**(3), 334–344 (2016)
- Hogan, N.: Impedance control: an approach to manipulation-part i: theory; part ii implementation; part iii: applications. *Trans ASME J. Dyn. Syst. Meas. Control* **107**(2), 1–24 (1981)
- Jiang, Y., Jiang, Z.P.: Computational adaptive optimal control for continuous-time linear systems with completely unknown dynamics. *Automatica* **48**(10), 2699–2704 (2012)

- Jiang, Y., Jiang, Z.: Global adaptive dynamic programming for continuous-time nonlinear systems. *IEEE Trans. Autom. Control* **60**(11), 2917–2929 (2015)
- Landi, C.T., Ferraguti, F., Sabattini, L., Secchi, C., Fantuzzi, C.: Admittance control parameter adaptation for physical human-robot interaction. In: 2017 IEEE International Conference on Robotics and Automation (ICRA), pp. 2911–2916 (2017)
- Liu, D., Wang, D., Wang, F., Li, H., Yang, X.: Neural-network-based online hjb solution for optimal robust guaranteed cost control of continuous-time uncertain nonlinear systems. *IEEE Trans. Cybern.* **44**(12), 2834–2847 (2014)
- Liu, D., Wei, Q., Wang, D., Yang, X., Li, H.: Adaptive dynamic programming with applications in optimal control. Springer, New York (2017)
- Love, L., Book, W.: Force reflecting teleoperation with adaptive impedance control. *IEEE Trans. Syst. Man Cybern. Part B: Cybern.* **34**, 159–165 (2004)
- Lyshevski, S.E.: Optimal control of nonlinear continuous-time systems: design of bounded controllers via generalized nonquadratic functionals. pp. 205–209 (1998)
- Parra-Vega, V., Arimoto, S., Yun-Hui, L., Hirzinger, G., Akella, P.: Dynamic sliding pid control for tracking of robot manipulators: theory and experiments. *IEEE Trans. Robot. Autom.* **19**(6), 967–976 (2003)
- Peng, G., Yang, C., He, W., Chen, C.L.P.: Force sensorless admittance control with neural learning for robots with actuator saturation. *IEEE Trans. Ind. Electron.* **67**(4), 3138–3148 (2020)
- Raibert, H.M., Craig, J.J., et al.: Hybrid position/force control of manipulators. *J. Dyn. Syst. Meas. Control* **103**(2), 126–133 (1981)
- Siciliano, B.: A closed-loop inverse kinematic scheme for on-line joint-based robot control. *Robotica* **8**, 231–243 (1990)
- Stanisic, R.Z., Fernandez, N.V.: Adjusting the parameters of the mechanical impedance for velocity, impact and force control. *Robotica* **30**(4), 583597 (2012)
- Tsujii, T., Ito, K., Morasso, P.: Neural network learning of robot arm impedance in operational space. *IEEE Trans. Syst. Man Cybern. Part B Cybern.* **26**, 290–8 (1996)
- Uemura, M., Kawamura, S.: Resonance-based motion control method for multi-joint robot through combining stiffness adaptation and iterative learning control. pp. 1543–1548 (2009)
- Wang, D., Liu, D., Mu, C., Zhang, Y.: Neural network learning and robust stabilization of nonlinear systems with dynamic uncertainties. *IEEE Trans. Neural Netw. Learn. Syst.* **29**(4), 1342–1351 (2018)
- Wen, C., Zhou, J., Liu, Z., Su, H.: Robust adaptive control of uncertain nonlinear systems in the presence of input saturation and external disturbance. *IEEE Trans. Autom. Control* **56**(7), 1672–1678 (2011)
- Wenzhi, G., Selmic, R.R.: Neural network control of a class of nonlinear systems with actuator saturation. *IEEE Trans Neural Netw* **17**(1), 147–156 (2006)
- Werbos, P.: Approximate dynamic programming for real-time control and neural modeling. Van Nostrand Reinhold, New York (1992)
- Yang, X., Liu, D., Huang, Y.: Neural-network-based online optimal control for uncertain non-linear continuous-time systems with control constraints. *IET Control Theory Appl.* **7**(17), 2037–2047 (2013)
- Yang, C., Peng, G., Li, Y., Cui, R., Cheng, L., Li, Z.: Neural networks enhanced adaptive admittance control of optimized robotenvironment interaction. *IEEE Trans. Cybern.* **49**(7), 2568–2579 (2019)
- Yang, C., Teng, T., Xu, B., Li, Z., Na, J., Su, C.Y.: Global adaptive tracking control of robot manipulators using neural networks with finite-time learning convergence. *Int. J. Control Autom. Syst.* **15**(4), 1916–1924 (2017)
- Yao, B., Zhou, Z., Wang, L., Xu, W., Liu, Q., Liu, A.: Sensorless and adaptive admittance control of industrial robot in physical human-robot interaction. *Robot. Comput.-Integr. Manuf.* **51**, 158–168 (2018)

- Zhan, H., Huang, D., Chen, Z., Wang, M., Yang, C.: Adaptive dynamic programming-based controller with admittance adaptation for robotenvironment interaction. *Int. J. Adv. Robot. Syst.* **17**(3), (2020)
- Zhang, S., Dong, Y., Ouyang, Y., Yin, Z., Peng, K.: Adaptive neural control for robotic manipulators with output constraints and uncertainties. *IEEE Trans. Neural Netw. Learn. Syst.* **29**(11), 5554–5564 (2018)
- Zhao, B., Jia, L., Xia, H., Li, Y.: Adaptive dynamic programming-based stabilization of nonlinear systems with unknown actuator saturation. *Nonlinear Dyn.* **93**(4), 2089–2103 (2018)

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Hong Zhan received the M.S. degree in automation from South China University of Technology, Guangzhou, China, in 2012. Then, she works in School of Automation Science and Engineering in South China University of Technology, where she is currently pursuing the Ph.D. degree. Her research interests include robotics, intelligent control and human-robot interaction.

Dianye Huang received the B.Eng. and M.Eng. degrees in automation from South China University of Technology, Guangzhou, China, in 2017 and 2020, respectively. His research interests include robotics, intelligent control and human-robot interaction.

Chenguang Yang received the Ph.D. degree in control engineering from the National University of Singapore, Singapore, in 2010 and performed postdoctoral research in human robotics at Imperial College London, London, UK from 2009 to 2010. 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.