International Journal of Advanced Robotic Systems
May-June 2020: 1–11
© The Author(s) 2020
Article reuse guidelines:
sagepub.com/journals-permissions
DOI: 10.1177/1729881420924610
journals.sagepub.com/home/arx
SAGE

Adaptive dynamic programming-based controller with admittance adaptation for robot–environment interaction

Hong Zhan[1], Dianye Huang[1], Zhaopeng Chen[2], Min Wang[1] and Chenguang Yang[3]

Abstract

The problem of optimal tracking control for robot–environment interaction is studied in this article. The environment is regarded as a linear system and an admittance control with iterative linear quadratic regulator method is obtained to guarantee the compliant behaviour. Meanwhile, an adaptive dynamic programming-based controller is proposed. Under adaptive dynamic programming frame, the critic network is performed with radial basis function neural network to approximate the optimal cost, and the neural network weight updating law is incorporated with an additional stabilizing term to eliminate the requirement for the initial admissible control. The stability of the system is proved by Lyapunov theorem. The simulation results demonstrate the effectiveness of the proposed control scheme.

Keywords

Adaptive dynamic programming, admittance adaptation, neural network, optimal control, robot–environment interaction

Date received: 17 July 2019; accepted: 7 April 2020

Topic: Robot Manipulation and Control
Topic Editor: Lino Marques
Associate Editor: Jian Huang

Introduction

Robot applications are becoming more and more widespread, such as rehabilitation therapy, assembly automation and surgery.[1–4] They can either work independently to accomplish tasks or cooperate with their human partners for certain tasks. In the actual application process, the robot will inevitably interact with the external environments.[5–7] Consequently, in recent years, interaction control between the robot and environment has attracted great concern and is considered to be greatly important.

In existing research, two main approaches are applied to achieve compliant behaviour of the robot, that is, hybrid position/force control and impedance control.[8,9] The first approach requires the position subspace and force subspace decomposition, task planning and control law switching in the execution process. Without considering the dynamic coupling of the environment and the robot, the accuracy of the hybrid position/force control cannot be guaranteed.[10] In contrast, the second approach aims to adjust the mechanical impedance to a target one, which will guarantee the robot to be complaint with the interaction force imposed by the external environment. Impedance control ensures the safety of the robot and the environment and it has been

[1] Key Lab of Autonomous Systems and Networked Control, Ministry of Education, School of Automation Science and Engineering, South China University of Technology, Guangzhou, China
[2] TAMS Group, Department of Informatics, University of Hamburg, Hamburg, D22527 Hamburg, Germany
[3] Bristol Robotics Laboratory, University of the West of England, Bristol, UK

Corresponding author:
Chenguang Yang, Bristol Robotics Laboratory, University of the West of England, Bristol BS16 1QY, UK.
Email: cyang@ieee.org

proved to be more feasible and has better robustness. According to the causality of the controller, impedance control has two implementation methods, one is named impedance control and the other is admittance control. In impedance control system, the interaction force can be estimated from the desired motion trajectory and impedance model, while in admittance control system, the reference trajectory is obtained from the measured environmental external force and the desired admittance model. Therefore, in this article, admittance control is adopted to solve the problem of robot–environment interaction control.

In admittance control system, force and the admittance model are two important parts. When robot–environment interaction exists, the force can be detected and measured by the sensors installed on the end-effector of the robot arm. But, how to derive optimal parameters of the admittance model is non-trivial. On the one hand, it is usually difficult to derive the desired admittance model because of the complexity of environmental dynamics; on the other hand, a fixed admittance model cannot satisfy all cases. Taking human–robot cooperation as an example, variable admittance control is necessary to ensure more efficient performance.[11] To solve these problems, iterative learning has been studied in robot intelligent control area. It has been investigated to obtain admittance parameters to adapt to unknown environment. The aim of this approach is to introduce human learning skills into the robot and improve control performance by repeating a task. Cohen and Flash[12] proposed an impedance learning control scheme using an associative search network to complete a wall-following work. Neural network (NN) is introduced into the impedance control to regulate the parameters.[13] However, the iterative learning method requires the robot to operate repeatedly, which brings inconvenience in practical process and is not feasible in many situations. Love and Book,[14] Uemura and Kawamura,[15] Gribovskaya et al.,[16] Stanisic and Fernández,[17] Landi et al.[18] and Yao et al.[19] have proposed to utilize adaptation approaches to address the problems stated above.

Robotic motion control is a challenging task as it is difficult to obtain accurate model concerning that the robot is a non-linear and highly coupled system. Proportional–integral–derivative (PID) control, NN control, adaptive control and other control methods have been applied to the robot system.[20–27] As a classical control method, PID control is employed to the robot system and can track the given reference trajectory well.[28] It is acknowledged that PID control has some advantages, such as simple structure and good robustness, but it is not easy to select suitable PID parameters if the controlled plant is complex. In addition, when dynamic uncertainties exist in the system, PID control cannot satisfy the performance requirements for the magnitude of overshoot, the rising and settling time and so on. NN has the fundamental characteristics of human brain and can simulate human behaviour for information processing, therefore it is widely used in the control field for unknown system identification. NN control can model the uncertain dynamics online to improve the system performance.[29] An admittance adaptation method and the NN-based controller are applied into the robot system.[30]

Tracking control is a significant research issue in the domain of robot intelligent control. For a controlled system, stability is just the minimum requirement. Optimal control needs to be considered, that is, it is required to design an optimal tracking controller, which could ensure system stability of the robot while minimizing the cost function. Werbos[31] proposed adaptive dynamic programming (ADP) strategy and it is considered to be an effective approach to resolve the optimal control problem.[32] The key of ADP method is to find a solution of Hamilton–Jacobi–Bellman (HJB) equation. However, because it is a partial differential equation, when the controlled system is nonlinear but not linear, its analytical solution will be very difficult to obtain, or even impossible. To solve the above problem, policy iterative is considered as an effective method to find the approximate solution, which requires initial stability control.[33] However, in practical process, the initial admissible control is usually very difficult to satisfy. Then, NN is introduced to derive an approximate solution of the HJB equation. The approximate solution is obtained by NN-based method, meanwhile the requirement of initial stability is eliminated with the incorporation of an additional term.[34,35]

Yang et al.[30] paid attention to the robot–environment interaction control, but did not consider the optimization problem. However, for the robot, how to perform path tracking optimization and minimize the cost function is very important. Based on the above discussion, the optimal tracking control problem for robot–environment interaction is studied in this article. Moreover, the admittance control and ADP approach are adopted to improve the system performance. The contributions of this article are listed below:

1. The environment with unknown dynamics is modelled as a linear system. An admittance adaptation method with iterative linear–quadratic regulator (LQR) is obtained to achieve a compliant behaviour.
2. ADP approach is introduced into the robot system to solve the optimal tracking problem. The critic network with radial basis function (RBF) is developed to approximate the minimum cost function. In addition, to eliminate the requirement for initial admissible control, a stabilizing term is incorporated into NN weight updating law.

The rest of this article is arranged as follows. Firstly, the robot and environment systems and control objectives are described. Next, the control scheme including admittance adaptation and optimal control using ADP is developed. Then, simulation studies are given. Finally, the conclusion is drawn.

Preliminaries and problem formulation

Robot dynamics

The n-link robot manipulator dynamics is showed as the following Lagrangian form

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

where $q = [q_1, q_2, \ldots, q_n]^{\mathrm{T}} \in \mathbb{R}^n$, $\dot{q} = [\dot{q}_1, \dot{q}_2, \ldots, \dot{q}_n]^{\mathrm{T}} \in \mathbb{R}^n$ and $\ddot{q} = [\ddot{q}_1, \ddot{q}_2, \ldots, \ddot{q}_n]^{\mathrm{T}} \in \mathbb{R}^n$ represent the robot position vector, velocity vector and acceleration vector in joint space, respectively. $\mu \in \mathbb{R}^n$ is the joint torque, while $M(q) \in \mathbb{R}^{n\times n}$, $C(q,\dot{q}) \in \mathbb{R}^{n\times n}$ and $G(q) \in \mathbb{R}^n$ are known matrices and denote the inertial matrix, Coriolis/centrifugal matrix and gravity vector, respectively. For convenience, M, C and G denote the known matrices $M(q)$, $C(q,\dot{q})$ and $G(q)$ in the following section, respectively.

Define the reference trajectory as $q_r \in \mathbb{R}^n$, and the tracking error $q_e \in \mathbb{R}^n$ is shown as follows

$$q_e = q - q_r \tag{2}$$

Then, the first and second time derivative of q_e are given below

$$\begin{aligned} \dot{q}_e &= \dot{q} - \dot{q}_r \\ \ddot{q}_e &= \ddot{q} - \ddot{q}_r \end{aligned} \tag{3}$$

We define the sliding motion surface ξ as follows

$$\xi = \Lambda q_e + \dot{q}_e \tag{4}$$

where $\Lambda \in R^{n\times n}$ is a constant positive matrix. According to equations (2) to (4), we can get

$$\begin{aligned} \dot{q} = &\ \xi - \Lambda q_e + \dot{q}_r \\ \ddot{q} = &\ \dot{\xi} - \Lambda q_e + \ddot{q}_r \end{aligned} \tag{5}$$

Substituting equation (5) into equation (1), the error dynamics is obtained as follows

$$\begin{aligned} \dot{\xi} = &-M^{-1}C(\xi - \Lambda q_e + \dot{q}_r) - M^{-1}G \\ &-\ddot{q}_r + \Lambda \dot{q}_e + M^{-1}\mu \end{aligned} \tag{6}$$

Then, the following system is obtained

$$\dot{\xi} = f(\xi) + g(\xi)\mu \tag{7}$$

The non-linear functions $f : \mathbb{R}^n \to \mathbb{R}^n$ and $g : \mathbb{R}^n \to \mathbb{R}^{n\times n}$ in equation (7) are specified by

$$\begin{aligned} &f(\xi) = -M^{-1}C(\xi - \Lambda q_e + \dot{q}_r) - M^{-1}G - \ddot{q}_r + \Lambda \dot{q}_e \\ &g(\xi) = M^{-1} \end{aligned} \tag{8}$$

Environment dynamics

It is assumed that the dynamics of environmental interaction force subject to the equation given below

$$C_E\dot{x} + G_E x = -F \tag{9}$$

where C_E and G_E represent the unknown damping and stiffness of the environment, respectively. F denotes the interaction force and can be detected and measured by a force sensor. x is the end-effector position in Cartesian space and the corresponding desired trajectory x_d is defined as

$$\dot{x}_d = U_d x_d \tag{10}$$

where $U \in \mathbb{R}^{m\times m}$ is a known matrix. Subsequently, we define $\eta = [x, x_d]^{\mathrm{T}}$. Thus, combining equation (9) with equation (10), dynamics of the unknown environment and the desired trajectory are generated by

$$\begin{aligned} \dot{\eta} &= \begin{bmatrix} -C_E^{-1}G_E & \mathbf{0} \\ \mathbf{0} & U_d \end{bmatrix}\eta + \begin{bmatrix} -C_E^{-1} \\ \mathbf{0} \end{bmatrix}F \\ &= A_e\eta + B_eF \end{aligned} \tag{11}$$

If we take equation (11) as a linear system with F as its control input and η as its states to be controlled, this equation relates x with x_d via the optimal feedback control law $F = -K_e\eta$ whose aim is to minimize the cost function

$$\Gamma_1 = \int_0^\infty \left(x_e^{\mathrm{T}} Q_{E1} x_e + F^{\mathrm{T}} R_E F\right) \mathrm{d}t \tag{12}$$

This cost function also indicates that our motivation of modifying a desired trajectory x_d is to balance the contact force F with the tracking error $x_e =: x - x_d$. And this balance can be tuned via the user-defined Q_{E1} and R_E.

In this section, the robot and environment dynamics are modelled. Then, we will design a control strategy to achieve the compliant behaviour and optimal tracking control in case the robot interacts with the environment.

Control scheme

A control scheme consisting of three parts as shown in Figure 1 including an optimal trajectory modifier using admittance control, a closed-loop inverse kinematics (CLIK) solver and a trajectory tracking controller based on ADP technique is designed in this section.

Trajectory modification using admittance control

The solution to equation (12) is an analogy with the LQR problem. It can be rewritten as

$$\begin{aligned} \Gamma &= \int_0^\infty \left(\eta^{\mathrm{T}} Q_E \eta + F^{\mathrm{T}} R_E F\right) \mathrm{d}t \\ Q_E &= \begin{bmatrix} Q_{E1} & -Q_{E1}U_d \\ -U_d^{\mathrm{T}} Q_{E1} & U_d^{\mathrm{T}} Q_{E1} U_d \end{bmatrix} \end{aligned} \tag{13}$$

whose system counterpart is consistent with equation (11).

In this subsection, an algorithm proposed by Jiang and Jiang[36] is adopted to solve the algebraic Riccati equation (ARE) in equation (14) with unknown environment parameters C_E, G_E to derive the feedback gain K_e

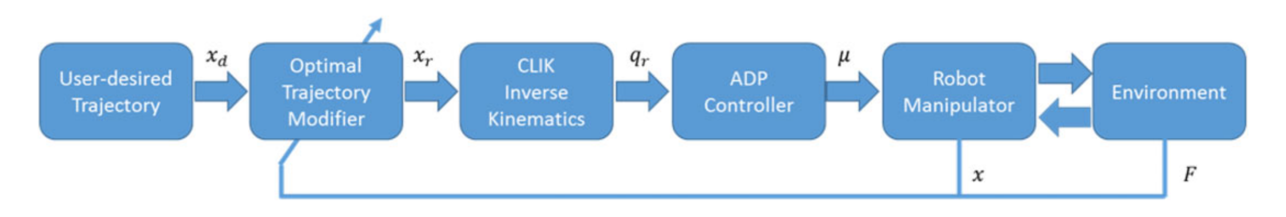

Figure 1. An illustration of the proposed control scheme.

$$\begin{aligned} &PA_e + A_e^{\mathrm{T}}P + Q_E - PB_eR_E^{-1}B_e^{\mathrm{T}}P = 0 \\ &K_e = -R_E^{-1}B_e^{\mathrm{T}}P \end{aligned} \tag{14}$$

Some notations are outlined here. n, m and d are the length of η, F and the sample times integer, respectively. The sampled signal together with the historical ones comprising the matrix as follows

$$\begin{aligned} \hat{p} &= [p_{11}, 2p_{12}, \ldots, 2p_{1n}, p_{22}, 2p_{23}, \ldots, p_{nn}]^{\mathrm{T}} \\ \bar{\eta} &= [\eta_1^2, \eta_1\eta_2, \ldots, \eta_1\eta_n, \eta_2^2, \eta_2\eta_3, \ldots, \eta_n^2]^{\mathrm{T}} \\ d\bar{\eta} &= [\bar{\eta}(t_1) - \bar{\eta}(t_0), \bar{\eta}(t_2) - \bar{\eta}(t_1), \ldots, \bar{\eta}(t_d) - \bar{\eta}(t_{d-1})]^{\mathrm{T}} \\ I_\eta^\eta &= \left[\int_{t_0}^{t_1} \eta \otimes \eta \,\mathrm{d}t, \int_{t_1}^{t_2} \eta \otimes \eta \,\mathrm{d}t, \ldots, \int_{t_{d-1}}^{t_d} \eta \otimes \eta \,\mathrm{d}t\right]^{\mathrm{T}} \\ I_F^\eta &= \left[\int_{t_0}^{t_1} \eta \otimes f \,\mathrm{d}t, \int_{t_1}^{t_2} \eta \otimes F \,\mathrm{d}t, \ldots, \int_{t_{d-1}}^{t_d} \eta \otimes F \,\mathrm{d}t\right]^{\mathrm{T}} \end{aligned} \tag{15}$$

where $p \in \mathbb{R}^{\frac{1}{2}n(n+1)}$, $\bar{\eta} \in \mathbb{R}^{\frac{1}{2}n(n-1)}$, $d\bar{\eta} \in \mathbb{R}^{d\times\frac{1}{2}n(n-1)}$, $I_\eta^\eta \in \mathbb{R}^{d\times n^2}$, $I_F^\eta \in \mathbb{R}^{d\times nm}$ and $\otimes$ stand for the Kronecker product, and p_{ij} and η_i denote entries of P and η, respectively

$$\mathrm{rank}\left(\left[I_\eta^\eta, I_F^\eta\right]\right) = \frac{n(n+1)}{2} + nm \tag{16}$$

When the number of sampled data is large enough and the rank condition in equation (16) is satisfied, the algorithm can solve K_e by iteratively calculating equation (17) until $\hat{p}^{(k)}$ converge to an acceptable range ε, that is, $||\hat{p}^{(k)} - \hat{p}^{(k-1)}|| < \varepsilon$ with $\| * \|$ denoting the 2-norm of $*$

$$\begin{aligned} Q_E^{(k)} &= Q_E + K_e^{(k)\mathrm{T}}R_EK_e^{(k)} \\ \Theta^{(k)} &= \left[d\bar{\eta}, -2I_\eta^\eta\left(I_n \otimes K_e^{(k)\mathrm{T}}R_E\right) - 2I_F^\eta(I_n \otimes R_E)\right] \\ \Xi^{(k)} &= -I_\eta^\eta \mathrm{vec}\left(Q_E^{(k)}\right) \\ \begin{bmatrix} \hat{p}^{(k)} \\ \mathrm{vec}\left(K_e^{(k+1)}\right) \end{bmatrix} &= \left(\Theta^{(k)\mathrm{T}}\Theta^{(k)}\right)^{-1}\Theta^{(k)\mathrm{T}}\Xi^{(k)} \end{aligned} \tag{17}$$

where the superscript (k) denotes the index of the iteration, vec$(*)$ denotes the column vectorization of $*$ and $I_n \in \mathbb{R}^{n\times n}$ is an identity matrix.

Once the optimal feedback gain K_e is obtained, we can use it to modify x_d. Formulations are given as below

$$F = -K_e\eta = -[K_{e1} \quad K_{e2}]\begin{bmatrix} x \\ x_d \end{bmatrix} \tag{18}$$

where K_{e1} and K_{e2} are compatible matrix from K_e. Finally, the modified trajectory x_r to be tracked is calculated, which is equivalent to the x in equation (18)

$$x_r = -K_{e1}^{-1}F - K_{e1}^{-1}K_{e2}x_d \tag{19}$$

Inverse kinematics using CLIK

The CLIK algorithm is employed to resolve the Cartesian reference trajectory x_r into the one q_r in joint space.[37] Let the solution error $e := \kappa(q_r) - x_r$ where $\kappa(*)$ denotes the forward kinematics and e is given by

$$\dot{e} = -K_fe \tag{20}$$

where K_f is a positive user-defined matrix that decides the convergent rate of e. Expanding the above equations and combining with $\dot{x} = J_{\mathrm{co}}\dot{q}$ and $J_{\mathrm{co}} = \partial\kappa(q)/\partial q$, the following equation holds

$$\dot{q}_r = J_{\mathrm{co}}^{\dagger}(\dot{x}_r - K_f(\kappa(q_r) - x_r)) \tag{21}$$

integrating of which yields the CLIK method

$$q_r = \int_0^t \left(J^{\dagger}\dot{x}_r - J_{\mathrm{co}}^{\dagger}K_f(\kappa(q_r) - x_r)\right)\mathrm{d}t \tag{22}$$

where $q(0) = \kappa^{-1}(x_r(0))$, $J_{\mathrm{co}}^{\dagger} = J_{\mathrm{co}}^{\mathrm{T}}\left(J_{\mathrm{co}}J_{\mathrm{co}}^{\mathrm{T}} + \sigma I_n\right)^{-1}$, and $\sigma \in \mathbb{R}$ is introduced to avoid the singularity problem which is recommended to be assigned small enough for improving the solution accuracy.

Optimal control using ADP

As mentioned in the Introduction section, it is very important to optimize the trajectory tracking while minimizing the design cost for robots. On the basis of optimal theory, the optimal control of the system (7) can be derived by solving the HJB equation in the frame of ADP. Consequently, in this subsection, our target is to find such an optimal control μ.

Assume that the functions $f(\xi)$ and $g(\xi)$ are Lipschitz continuous in $\mathbb{R}^{2n}$ and system (7) is controllable, then the optimal control μ^* should minimize the cost function which is expressed as

$$J(\xi(t)) = \int_t^{\infty} [\Phi(\xi(\tau)) + U(\xi(\tau), \mu(\xi(\tau)))]\mathrm{d}\tau \tag{23}$$

where $\Phi(\xi(\tau)) = \xi(\tau)^T Q \xi(\tau)$, $U(\xi(\tau), \mu(\xi(\tau))) = \mu(\xi(\tau))^T R \mu(\xi(\tau))$, $Q \in \mathbb{R}^{n \times n}$ and $R \in \mathbb{R}^{n \times n}$ are symmetric positive definite matrices. For robot system (7), the optimal control μ^* should not only guarantee system stability but also can make the cost function finite, that is, the control law should be in the admissible control set which defined as χ. Additionally, for any admissible control law $\mu \in \chi$, if $J(\xi)$ given in equation (23) is continuously differentiable, we will have the non-linear Lyapunov equation which is an infinitesimal version of equation (23) is shown as follows with $J(0) = 0$

$$\begin{aligned} 0 = & \ \Phi(\xi(\tau)) + U(\xi(\tau), \mu(\xi(\tau))) \\ & + (\nabla J(\xi))^{\mathrm{T}}(f(\xi) + g(\xi)\mu(\xi)) \end{aligned} \tag{24}$$

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

Then, the Hamiltonian function and the optimal cost function of robot system (7) are defined as below

$$\begin{aligned} H(\xi, \mu(\xi), \nabla J(\xi)) = & \ \Phi(\xi(\tau)) + U(\xi(\tau), \mu(\xi(\tau))) \\ & + (\nabla J(\xi))^{\mathrm{T}}(f(\xi) + g(\xi)\mu(\xi)) \end{aligned} \tag{25}$$

$$J(\xi)^* = \min_{\mu \in \chi} \int_t^{\infty} [\Phi(\xi(\tau)) + U(\xi(\tau), \mu(\xi(\tau)))]\mathrm{d}\tau \tag{26}$$

We can obtain the HJB equation shown as

$$0 = \min_{\mu \in \chi} H(\xi, \mu(\xi), \nabla J^*(\xi)) \tag{27}$$

Suppose that the minimum value on the right side of formula (27) exists and also is unique, from $\frac{\partial H(\xi, \mu(\xi), \nabla J^*(\xi))}{\partial \mu} = 0$, then the following optimal control $\mu^*(\xi)$ can be derived as

$$\mu^*(\xi) = -\frac{1}{2} R^{-1} g^{\mathrm{T}}(\xi) \nabla J^*(\xi) \tag{28}$$

Substituting the optimal control law (28) into equation (24) yields another form of HJB equation with respect to $\nabla J^*(\xi)$ is obtained as

$$H(\xi, \mu^*(\xi), \nabla J^*(\xi)) = 0 \tag{29}$$

Inspired by Liu et al.,[34] we know that if the optimal function $J^*(\xi)$ is assumed to be continuously differentiable, $J^*(\xi)$ can be rebuilded by RBFNN which can be shown as below

$$J^*(\xi) = w^{\mathrm{T}} S(\xi) + \varepsilon(\xi) \tag{30}$$

where $w \in \mathbb{R}^l$ represents the ideal constant weight, $S : \mathbb{R}^{2n} \to \mathbb{R}^l$ denotes the activation function, l denotes the node number in the hidden layer and $\varepsilon(\xi)$ denotes the unknown approximation error of NN. Then, the derivation of equation (30) involving ξ is derived as

$$\nabla J^*(\xi) = (\nabla S(\xi))^{\mathrm{T}} w + \nabla \varepsilon(\xi) \tag{31}$$

From equations (28) and (31), the following μ^* can be obtained as

$$\mu^*(\xi) = -\frac{1}{2} R^{-1} g^{\mathrm{T}}(\xi)((\nabla S(\xi))^{\mathrm{T}} w + \nabla \varepsilon(\xi)) \tag{32}$$

Then, substituting equations (31) and (32) into equation (29), we have

$$\begin{aligned} H^*(\xi, \mu^*(\xi), \nabla J^*(\xi)) = & \ \Phi(\xi) + w^{\mathrm{T}} \nabla S(\xi) f(\xi) \\ & - \frac{1}{4} w^{\mathrm{T}} \nabla S(\xi) D \nabla S(\xi)^{\mathrm{T}} w \\ & + e_c = 0 \end{aligned} \tag{33}$$

where

$$e_c = (\nabla \varepsilon(\xi))^{\mathrm{T}}(f(\xi) + g(\xi)\mu^*(\xi)) \tag{34}$$

$$D = g(\xi) R^{-1} g(\xi)^{\mathrm{T}} \tag{35}$$

In fact, the ideal weight w and $J^*(\xi)$ in equation (30) are unknown, then the estimate weight and optimal cost function, respectively, denoted as $\hat{w}$ and $J^*(\xi)$ can be obtained by the constructed critic NN. Therefore, the approximate optimal cost $J^*(\xi)$ is given as below

$$\hat{J}(\xi) = \hat{w}^{\mathrm{T}} S(\xi) \tag{36}$$

Then, the derivative of equation (36) is

$$\nabla \hat{J}(\xi) = (\nabla S(\xi))^{\mathrm{T}} \hat{w} \tag{37}$$

Based on equations (28) and (37), the approximate optimal control is obtained as

$$\hat{\mu}(\xi) = -\frac{1}{2} R^{-1} g^{\mathrm{T}}(\xi)(\nabla S(\xi))^{\mathrm{T}} \hat{w} \tag{38}$$

Similarly, applying equations (25), (37) and (38), the approximate Hamiltonian function $\hat{H}(\xi, \hat{\mu}(\xi), \nabla \hat{J}(\xi))$ can be derived as

$$\begin{aligned} \hat{H}(\xi, \hat{\mu}(\xi), \nabla \hat{J}(\xi)) = & \ \Phi(\xi) + \hat{w}^{\mathrm{T}} \nabla S(\xi) f(\xi) \\ & - \frac{1}{4} \hat{w}^{\mathrm{T}} \nabla S(\xi) D (\nabla S(\xi))^{\mathrm{T}} \hat{w} \end{aligned} \tag{39}$$

Define e_H as the error between H^* and $\hat{H}$, $\tilde{w}$ as the approximate NN weight error, then they are shown as below

$$\begin{aligned} e_H = & \ \hat{H}(\xi, \hat{\mu}(\xi), \nabla \hat{J}(\xi)) \\ & - H^*(\xi, \mu^*(\xi), \nabla J^*(\xi)) \end{aligned} \tag{40}$$

$$\tilde{w} = w - \hat{w} \tag{41}$$

Table 1. Parameters of the robot manipulator.

Parameters	Values
l_1	0.50 m
l_{c1}	0.25 m
l_2	0.50 m
l_{c2}	0.25 m
m_1	5 kg
m_2	5 kg

According to equations (33), (39) and (41), e_H in equation (40) can be described as

$$\begin{aligned} e_H &= \hat{H}(\xi, \hat{\mu}(\xi), \nabla\hat{J}(\xi)) \\ &= -\tilde{w}^{\mathrm{T}}\nabla S(\xi)f(\xi) + \frac{1}{2}\tilde{w}^{\mathrm{T}}\nabla S(\xi)D(\nabla S(\xi))^{\mathrm{T}}w \\ &\quad -\frac{1}{4}\tilde{w}^{\mathrm{T}}\nabla S(\xi)D(\nabla S(\xi))^{\mathrm{T}}\tilde{w} - e_c \end{aligned} \tag{42}$$

To train RBFNN, an appropriate weight updating law $\hat{w}$ should be designed to both minimize the objective function $E = \frac{1}{2}e_H^2$ and ensure the approximate optimal weight $\hat{w}$ converge to the ideal weight w. To eliminate the requirement for the initial admissible control law, the weight $\hat{w}$ is tuned according to the standard gradient descent algorithm with an additional stabilizing term. The weight updating law is given as

$$\begin{aligned} \dot{\hat{w}} &= -(1-h)\alpha_H\left(\frac{\partial E}{\partial \hat{w}}\right) \\ &\quad + \frac{1}{2}h\alpha_c\left(\frac{\partial(\nabla J_s(\xi))^{\mathrm{T}}(f(\xi)+g(\xi)\hat{\mu})}{\partial\hat{w}}\right) \\ &= -(1-h)\alpha_H\left(\frac{\partial E}{\partial \hat{w}}\right) + \frac{1}{2}h\alpha_c\nabla S(\xi)D\nabla J_s(\xi) \end{aligned} \tag{43}$$

$$\begin{aligned} \frac{\partial E}{\partial \hat{w}} &= e_H\frac{\partial e_H}{\partial \hat{w}} \\ &= \hat{H}(\xi, \hat{\mu}^*(\xi), \nabla\hat{J}(\xi))\frac{\partial \hat{H}}{\partial \hat{w}} \\ &= [\nabla S(\xi)f(\xi) - \frac{1}{2}\nabla S(\xi)D\nabla S(\xi)^{\mathrm{T}}\hat{w}][\Phi(\xi(\tau)) \\ &\quad +\hat{w}^{\mathrm{T}}\nabla S(\xi)f(\xi) - \frac{1}{4}\hat{w}^{\mathrm{T}}\nabla S(\xi)D\nabla S(\xi)^{\mathrm{T}}\hat{w}] \end{aligned} \tag{44}$$

where α_H and α_c are the basic learning rate of the standard gradient descent algorithm and the learning rate of the stabilizing term, respectively. h is defined as follows

$$h = \begin{cases} 0, & \text{if}(\nabla J_s(\xi))^{\mathrm{T}}(f(\xi)+g(\xi)\hat{\mu}) < 0 \\ 1, & \text{else} \end{cases} \tag{45}$$

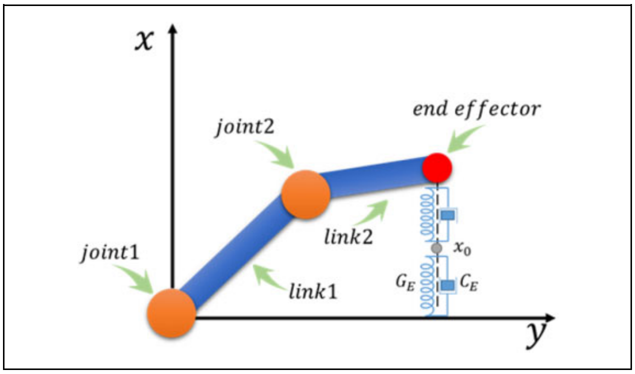

Figure 2. Settings of the numerical simulation.

where $J_s(\xi)$ is selected as a Lyapunov function candidate which is continuously differentiable. And assume that a positive definite matrix N exists, then the following equation is satisfied

$$\begin{aligned} \dot{J}_s(\xi) &= (\nabla J_s(\xi))^{\mathrm{T}}(f(\xi)+g(\xi)\mu^*) \\ &= -(\nabla J_s(\xi))^{\mathrm{T}}N\nabla J_s(\xi) < 0 \end{aligned} \tag{46}$$

It should be noted that $J_s(\xi)$ is a polynomial with the state variable and can be chosen appropriately, such as the form $J_s(\xi) = \frac{1}{2}\xi^{\mathrm{T}}\xi$.

Stability analysis

In this subsection, we will analyse the stability of the system and give the detailed proof that the approximate error $\tilde{w}$ of the NN weight and the state ξ are convergent.

Theorem 1. Consider the robot system (7) with approximate optimal control (38) and the NN weight updating law (43), then it is concluded that the approximate error $\tilde{w}$ of the NN weight and the state ξ are convergent.

Proof. See the Appendix.

Numerical simulation

Simulation settings

A two-degree-of-freedom (2-DOF) planar manipulator is adopted to verify the proposed control scheme. It is constructed by the robotics toolbox with parameters shown in Table 1.[38] The numerical simulation shown in Figure 2 runs on the MATLAB 2018a software where an ode3 solver is chosen with a fixed time step of 0.01 s, simulation time 20 s and other settings remain default. The initial joint position is q_0= $[0.08211, 1.897]^{\mathrm{T}}$ and the user-defined trajectory is $x_d(t) = [0.3\exp(-t), 0.5]^{\mathrm{T}}$. The environment dynamics is simulated as

$$-F = C_E\dot{x} + G_E(x - x_0) \tag{47}$$

where C_E, G_E and x_0 are chosen as diag$(0.1, 0.1)$, diag$(1.0, 1.0)$, 0.2, respectively, which are unknown

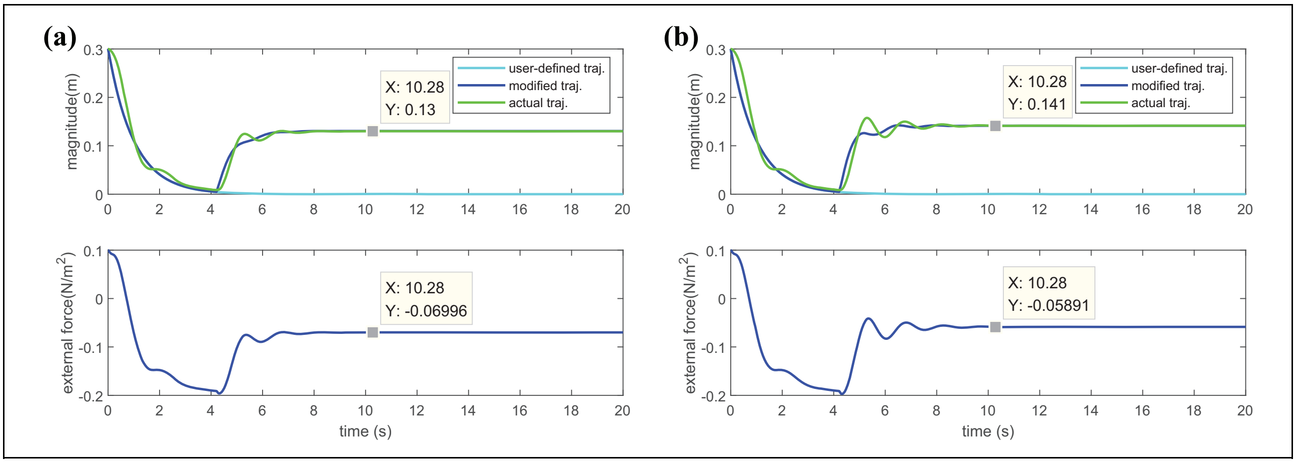

Figure 3. Simulation results of trajectory modification. (a) Case 1, $K_e = K_e^{\mathrm{pro}}$ and (b) Case 2, $K_e = K_e^{\mathrm{opt}}$.

during the simulation. For simplicity and without losing generality, only the trajectory along the x-axis is modified and interfered with the external forces.

For the proposed control scheme, parameters are set as below: to calculate the optimal trajectory in equation (13), $Q_{E1} = 1.0$, $R_E = 1.0$ and $U_d = -0.3$; the feedback gain in the inverse kinematics in equation (22), $K_f = 30$ and $\sigma = 1e-6$; as for the ADP controller, in equations (4), (23), (38) and (43), $\Lambda = \mathrm{diag}(5,5)$, $R = \mathrm{diag}(0.02,0.02)$, $Q = \mathrm{diag}(2.0,2.0)$, $\alpha_H = 0.5$ and $\alpha_c = 2.5$. Besides, an RBFNN is selected to approximate the cost function in equation (23), where $\hat{J} = \hat{w}^{\mathrm{T}}S(\xi)$, $S_i(\xi) = \exp(\| \xi - c_{\mathrm{rbf}} \| / \sigma_{\mathrm{rbf}}^2)$ with $\hat{w} \in \mathbb{R}^9$, $S(\xi) \in \mathbb{R}^9$, $\hat{w}(0) = 0$, $\sigma_{\mathrm{rbf}} = 0.55$, $c_{\mathrm{rbf}} \in [-0.2, 0.0, 0.2] \times [-0.2, 0.0, 0.2]$.

Simulation results

In this subsection, two cases will be compared to demonstrate the validity of the proposed scheme. Note that, the environment dynamics of the simulation is not totally consistent with that in equation (9), and x_0 is unknown. Therefore, two different K_e values are considered and examined. Case 1: the feedback gain $K_e^{\mathrm{pro}} = [-0.5367, 0.22840]$ acquired from the proposed scheme, which is different from Case 2: the ideal feedback gain $K_e^{\mathrm{opt}} = [-0.4142, 0.6604]$ obtained by calculating offline with the exact values of G_E and C_E (the unknown x_0 is ignored in this case). For fair comparison, in Case 2, the trajectory will be modified at the time as Case 1.

Simulation results are shown in Figures 3 to 6. Figure 3 shows the modification process of the user-defined trajectory along the x-axis of both cases. It is not until around 4.1 s that the rank condition in equation (16) is satisfied following that the trajectory starts being modified. During the transient process, it can be found that the modified trajectory of Case 2 has a slight oscillation, and this subsequently triggers larger tracking errors compared with Case 1. The steady state and force pair of Case 1 and Case 2 trajectories at 10.28 s are 0.13 m/−0.07 N and 0.14 m/−0.06 N, respectively, which is in line with the time series of the cost function in equation (12) of both cases as shown in Figure 4. From the figure, we can see that after the modification of trajectory, the cost function of Case 1 is smaller than that of Case 2, which implies that in this simulation settings where the actual existence of unknown x_0 cannot be neglected, the feedback gain obtained from the proposed scheme is more appropriate. Note that, due to the unknown x_0, the environment dynamics in equation (9) used for the designing of the

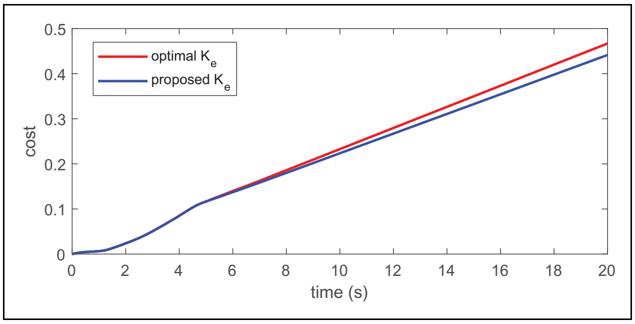

Figure 4. The time series of cost function in equation (12), $Q_{E1} = 1.0$ and $R_E = 1.0$.

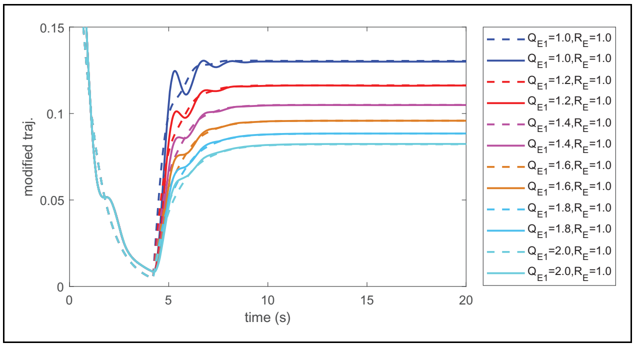

Figure 5. Modified trajectories corresponding to the different choice of Q_{E1} in equation (12).

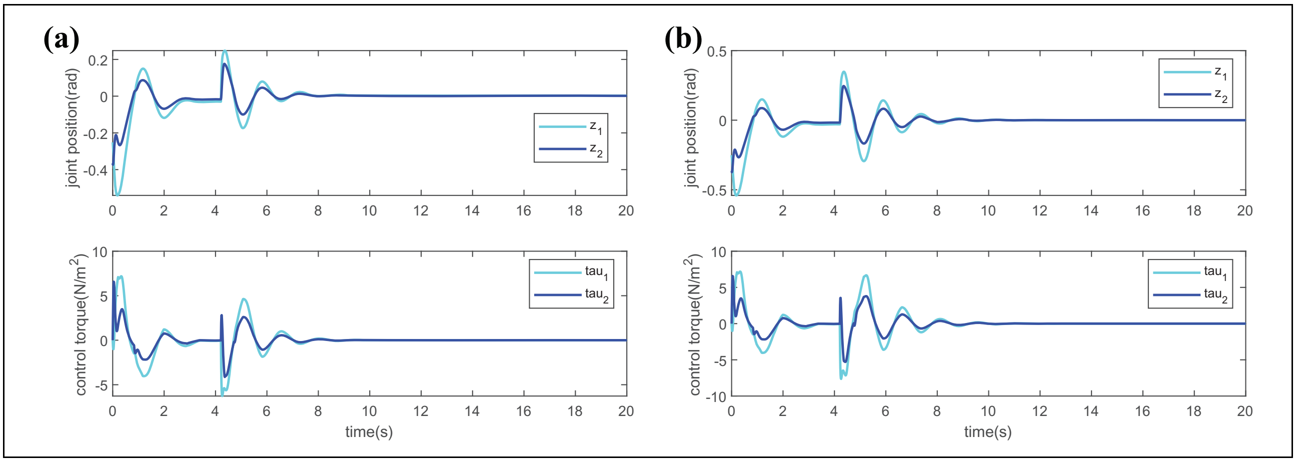

Figure 6. Simulation results of control performance. (a) Case 1, $K_e = K_e^{pro}$ and (b) Case 2, $K_e = K_e^{opt}$.

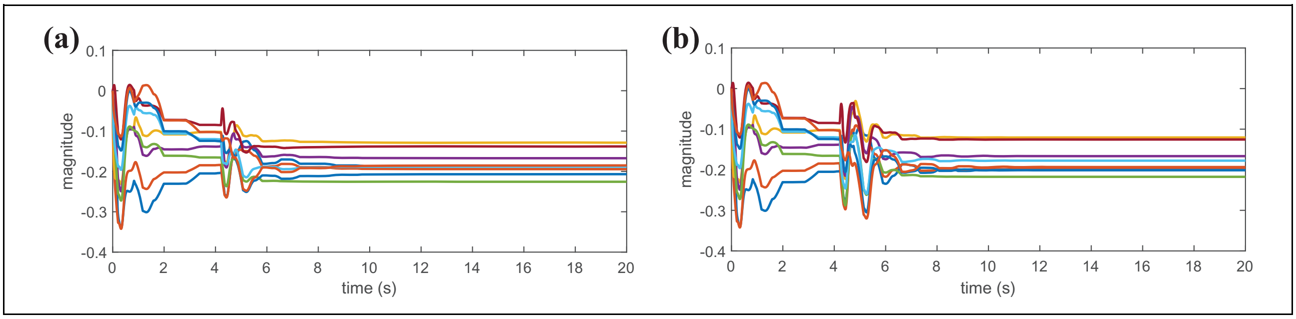

Figure 7. Weights of the RBFNN. (a) Case 1, $K_e = K_e^{pro}$ and (b) Case 2, $K_e = K_e^{opt}$.

Table 2. Feedback gains of different Q_{E1}.

	$Q_{E1} = 0.8$ $R_E = 1.0$	$Q_{E1} = 1.0$ $R_E = 1.0$	$Q_{E1} = 1.2$ $R_E = 1.0$	$Q_{E1} = 1.4$ $R_E = 1.0$	$Q_{E1} = 1.6$ $R_E = 1.0$	$Q_{E1} = 1.8$ $R_E = 1.0$	$Q_{E1} = 2.0$ $R_E = 1.0$
K_{e1}	−0.3625	−0.5367	−0.7205	−0.9055	−1.0870	−1.2620	−1.4300
K_{e2}	0.1090	0.2284	0.3606	0.4968	0.6317	0.7627	0.8884

trajectory modifier differentiates from that in equation (47) used for simulation. Therefore, under this situation, actually neither the K_e of Case 1 nor Case 2 is the optimal one. However, the proposed method still works and regards the dynamics in equation (47) as a linear one with an appropriate feedback gain. This has demonstrated the effectiveness of the proposed admittance control method.

Figures 5 to 7 are plotted for analysing the performance of the ADP-based controller. Figure 6 shows the control torques τ and sliding mode surface z of Case 1 and Case 2. On the whole, the proposed scheme tracks the both modified trajectory well, given that only nine neurons are used in the RBFNN, and the control torques are within the physical limitation. Besides, weights convergence can be observed in Figure 7. Note that, because of the introduced additional term ∇J_s, the initial admissible policy requirement is relaxed. Thus, in the simulation we choose the weights w to be zeros, without worrying about the control stability. This can be observed from Figure 6 that despite initial errors are large, they finally converge to zeros after some oscillations. Table 2 shows the feedback gain K_e calculated online using the proposed admittance control under the choices of different Q_{E1} in equation (12). Its corresponding reference trajectories are shown in Figure 5 where the dashed lines denote the reference trajectories after modification and the solid lines stand for the actual trajectories of the robot end-effector under the control of the proposed ADP controller. Obviously, as the Q_{E1} is selected larger, the reference trajectories tend to get closer to the user-desired trajectory, which is consistent with the fact that more cost is applied to the modified error X_e. Furthermore, although the reference trajectory varies, the proposed ADP controller is still eventually able to track the input signals with the same set of parameters. These also reflect the effectiveness of the proposed ADP controller and admittance control.

Conclusion

The optimal control of robots interacting between unknown environment was studied in this article. An ADP-based controller with admittance adaptation was proposed. The unknown environment was regarded as a linear system and a compliant behaviour was guaranteed by the admittance adaptation control. In addition, NN was introduced into ADP controller to ensure trajectory tracking of the robot with minimal cost. The stability of the robot system was proved and simulation studies demonstrated the effectiveness of the proposed control scheme.

Because of the complexity of the robot system, dynamic uncertainties and input constraints such as saturation and dead zone are very common in robot systems, which will not only affect system performance but also may lead to system instability.[24,39,40] Therefore, in the frame of ADP, the optimal control problem with dynamic uncertainties and input constraints will be considered in our future work.

Declaration of conflicting interests

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

Funding

The author(s) disclosed receipt of the following financial support for the research, authorship, and/or publication of this article: This work was partially supported by Engineering and Physical Sciences Research Council (EPSRC) under grant EP/S001913 and Shenzhen Science and Technology Plan Project [JSGG20180507183020876].

ORCID iD

Chenguang Yang https://orcid.org/0000-0001-5255-5559

References

1. He W, Li Z and Chen CLP. A survey of human-centered intelligent robots: issues and challenges. *IEEE/CAA J Autom Sinica* 2017; 4(4): 602–609.
2. Huang J, Cao Y, Xiong C, et al. An echo state Gaussian process-based nonlinear model predictive control for pneumatic muscle actuators. *IEEE Trans Autom Sci Eng* 2019; 16(3): 1071–1084.
3. Huang J, Wang Y and Fukuda T. Set-membership-based fault detection and isolation for robotic assembly of electrical connectors. *IEEE Trans Autom Sci Eng* 2018; 15(1): 160–171.
4. Chandrasekaran B and Conrad JM. Human–robot collaboration: a survey. In: *SoutheastCon 2015*, Fort Lauderdale, FL, USA, 9–12 April 2015, pp. 1–8. IEEE.
5. Mason MT. Compliance and force control for computer controlled manipulators. *IEEE Trans Syst Man Cybern* 1981; 11(6): 418–432.
6. Wang C, Li Y, Ge SS, et al. Reference adaptation for robots in physical interactions with unknown environments. *IEEE Trans Cybern* 2017; 47(11): 3504–3515.
7. Huang J, Ri M, Wu D, et al. Interval type-2 fuzzy logic modeling and control of a mobile two-wheeled inverted pendulum. *IEEE Trans Fuzzy Syst* 2018; 26(4): 2030–2038.
8. Raibert M and Craig J. Hybrid position/force control of manipulators. *Trans ASME J Dyn Syst Meas Control* 1981; 103(2): 126–133.
9. Hogan N. Impedance control: an approach to manipulation—Part I: theory; Part II: implementation; Part III: applications. *Trans ASME J Dyn Syst Meas Control* 1981; 107(2): 1–24.
10. Ott C, Mukherjee R and Nakamura Y. A hybrid system framework for unified impedance and admittance control. *J Intell Robot Syst* 2015; 78(3): 359–375.
11. Braun D, Petit F, Huber F, et al. Optimal torque and stiffness control in compliantly actuated robots. In: *2012 IEEERSJ international conference on intelligent robots and systems*, Vilamoura, Portugal, 7–12 October, 2012, pp. 2801–2808.
12. Cohen M and Flash T. Learning impedance parameters for robot control using an associative search network. *IEEE Trans Robot Autom* 1991; 7(3): 382–390.
13. Tsuji T, Ito K and Morasso PG. Neural network learning of robot arm impedance in operational space. *IEEE Trans Syst Man Cybern* 1996; 26(2): 290–298.
14. Love LJ and Book WJ. Force reflecting teleoperation with adaptive impedance control. *IEEE Trans Syst Man Cybern* 2004; 34(1): 159–165.
15. Uemura M and Kawamura S.Resonance-based motion control method for multi-joint robot through combining stiffness adaptation and iterative learning control. In: *2009 IEEE international conference on robotics and automation*, Kobe, Japan, 12–17 May 2009, pp. 1543–1548.
16. Gribovskaya E, Kheddar A and Billard A.Motion learning and adaptive impedance for robot control during physical interaction with humans. In: *2011 IEEE international conference on robotics and automation*, Shanghai, China, 9–13 May 2011, pp. 4326–4332.
17. Stanisic RZ and Fernández AV. Adjusting the parameters of the mechanical impedance for velocity, impact and force control. *Robotica* 2012; 30(4): 583–597.
18. Landi CT, Ferraguti F, Sabattini L, et al. Admittance control parameter adaptation for physical human-robot interaction. In: *2017 IEEE international conference on robotics and automation (ICRA)*, Singapore, Singapore, 29 May–3 June 2017, pp. 2911–2916.
19. Yao B, Zhou Z, Wang L, et al. Sensorless and adaptive admittance control of industrial robot in physical human–robot interaction. *Robot CIM-Int Manuf* 2018; 51: 158–168.
20. Cervantes I and Alvarez-Ramirez J. On the PID tracking control of robot manipulators. *Syst Control Lett* 2001; 42(1): 37–46.
21. Yu W and Rosen J. Neural PID control of robot manipulators with application to an upper limb exoskeleton. *IEEE Trans Cybern* 2013; 43(2): 673–684.

22. Huang J, Zhang M, Ri S, et al. High-order disturbance observer based sliding mode control for mobile wheeled inverted pendulum systems. *IEEE Trans Ind Electron* 2020; 67(3): 2030–2041.
23. Yang C, Teng T, Xu B, et al. Global adaptive tracking control of robot manipulators using neural networks with finite-time learning convergence. *Int J Control Autom Syst* 2017; 15(4): 1916–1924.
24. He W, Dong Y and Sun C. Adaptive neural impedance control of a robotic manipulator with input saturation. *IEEE Trans Syst Man Cyber: Syst* 2016; 46(3): 334–344.
25. Slotine JJE and Li W. On the adaptive control of robot manipulators. *Int J Robot Res* 1987; 6(3): 49–59.
26. Fukao T, Nakagawa H and Adachi N. Adaptive tracking control of a nonholonomic mobile robot. *IEEE Trans Robot Autom* 2000; 16(5): 609–615.
27. Huang J, Ri S, Fu T, et al. A disturbance observer based sliding mode control for a class of underactuated robotic system with mismatched uncertainties. *IEEE Trans Autom Control* 2019; 64(6): 2480–2487.
28. Parra-Vega V, Arimoto S, Liu YH, et al. Dynamic sliding PID control for tracking of robot manipulators: theory and experiments. *IEEE Trans Robot Autom* 2003; 19(6): 967–976.
29. Zhang S, Dong Y, Ouyang Y, et al. Adaptive neural control for robotic manipulators with output constraints and uncertainties. *IEEE Trans Neural Netw Learn* 2018; 29(11): 5554–5564.
30. Yang C, Peng G, Li Y, et al. Neural networks enhanced adaptive admittance control of optimized robot–environment interaction. *IEEE Trans Cybern* 2019; 49(7): 2568–2579.
31. Werbos P. *Approximate dynamic programming for real-time control and neural modeling*. New York: Van Nostrand Reinhold, 1992.
32. Wang F, Zhang H and Liu D. Adaptive dynamic programming: an introduction. *IEEE Comput Intell Mag* 2009; 4(2): 39–47.
33. Wang D, Liu D and Li H. Policy iteration algorithm for online design of robust control for a class of continuous-time nonlinear systems. *IEEE Trans Autom Sci Eng* 2014; 11(2): 627–632.
34. Liu D, Wang D, Wang FY, et al. Neural-network-based online HJB solution for optimal robust guaranteed cost control of continuous-time uncertain nonlinear systems. *IEEE Trans Cybern* 2014; 44(12): 2834–2847.
35. Wang D, Liu D, Mu C, et al. Neural network learning and robust stabilization of nonlinear systems with dynamic uncertainties. *IEEE Trans Neural Netw Learn Syst* 2018; 29(4): 1342–1351.
36. Jiang Y and Jiang ZP. Computational adaptive optimal control for continuous-time linear systems with completely unknown dynamics. *Automatica* 2012; 48(10): 2699–2704.
37. Siciliano B. A closed-loop inverse kinematic scheme for online joint-based robot control. *Robotica* 1990; 8: 231–243.
38. Corke P. *Robotics, vision and control*, Vol. 118. Berlin: Springer, 2017.
39. Sun N, Liang D, Wu Y, et al. Adaptive control for pneumatic artificial muscle systems with parametric uncertainties and unidirectional input constraints. *IEEE Trans Ind Inform* 2020; 16(2): 969–979.
40. Sun N, Wu Y, Fang Y, et al. Nonlinear antiswing control for crane systems with double-pendulum swing effects and uncertain parameters: design and experiments. *IEEE Trans Autom Sci Eng* 2018; 15(3): 1413–1422.

Appendix 1

Stability analysis

This appendix illustrates the stability of the proposed ADP-based controller. For the sake of brevity, the dependency of ξ will be omitted, for example, the notation $\nabla S(\xi)$ will be replaced with ∇S. Besides, define $\dot{\bar{\xi}}$ as the optimal time derivative system states corresponding to the optimal control, $\dot{\bar{\xi}} = (f + g\mu^*) = f - \frac{1}{2}D\nabla S^{\mathrm{T}}w$. Thus

$$\dot{\xi} = \dot{\bar{\xi}} + \frac{1}{2}D\nabla S^T\tilde{w} \tag{1A}$$

The Lyapunov candidate is selected as below

$$V = \frac{1}{\alpha_H}\tilde{w}^T\tilde{w} + \frac{\alpha_c}{2}\xi^T\xi \tag{1B}$$

Combining with equations (7), (38), (41) and (43), its time derivative is

$$\dot{V} = -\tilde{w}^{\mathrm{T}}\left(-(1-h)\left(\frac{\partial E}{\partial \hat{w}}\right) + \frac{1}{2}h\alpha_c\nabla SD\nabla J_s\right) + \alpha_c\nabla J_s^{\mathrm{T}}\dot{\xi} \tag{1C}$$

Case 1. $h = 1$, namely, $(\nabla J_s)^{\mathrm{T}}\left(f - \frac{1}{2}D\nabla S^{\mathrm{T}}\hat{w}\right) \geq 0$. Then, along with equations (46), (1A) and (1C) is equal to

$$\begin{aligned}\dot{V}|_{h=1} &= -\alpha_c\frac{1}{2}\tilde{w}^{\mathrm{T}}\nabla SD\nabla J_s + \alpha_c\nabla J_s^{\mathrm{T}}\left(\dot{\bar{\xi}} + \frac{1}{2}D\nabla S^{\mathrm{T}}\tilde{w}\right)\\ &= \alpha_c\nabla J_s^{\mathrm{T}}\dot{\bar{\xi}} = \alpha_c\dot{J}_s < 0\end{aligned} \tag{1D}$$

Case 2. $h = 0$, namely, $\nabla J_s^{\mathrm{T}}\dot{\xi} < 0$. In this case, according to the density property of real numbers, there exists a positive constant λ_J such that $\alpha_c\lambda_J \parallel \nabla J_s \parallel < -\alpha_c\nabla J_s^{\mathrm{T}}\dot{\xi}$. Equation (1C) can be rewritten as

$$\dot{V}|_{h=0} = \tilde{w}^{\mathrm{T}}\left(\frac{\partial E}{\partial \hat{w}}\right) + \alpha_c\nabla J_s^{\mathrm{T}}\dot{\xi} \tag{1E}$$

Equation (44) can also be presented along with equation (42)

$$\begin{aligned}\frac{\partial E}{\partial \hat{w}} &= e_H \frac{\partial e_H}{\partial \hat{w}} \\ &= \left(-\tilde{w}^{\mathrm{T}} \nabla S f + \frac{1}{2} \tilde{w}^{\mathrm{T}} \nabla S D \nabla S^{\mathrm{T}} w - \frac{1}{4} \tilde{w}^{\mathrm{T}} \nabla S D \nabla S^{\mathrm{T}} \tilde{w} - e_c\right) \nabla S \dot{\xi} \\ &= -\left(\tilde{w}^{\mathrm{T}} \nabla S \dot{\bar{\xi}} + \frac{1}{4} \parallel \tilde{w} \parallel_{\bar{D}}^2 + e_c\right) \nabla S \left(\dot{\bar{\xi}} + \frac{1}{2} D \nabla S^{\mathrm{T}} \tilde{w}\right)\end{aligned} \tag{1F}$$

where $\bar{D} := \nabla S D \nabla S^{\mathrm{T}}$ and $\parallel \xi \|_A = \sqrt{\xi^{\mathrm{T}} A \xi}$ denotes the norm of ξ weighting by a compatible matrix A. Substituting equation (1F) into equation (1E) yields

$$\begin{aligned}\dot{V} = &-(\tilde{w}^{\mathrm{T}} \nabla S \dot{\bar{\xi}} + \frac{1}{4} \parallel \tilde{w} \parallel_{\bar{D}}^2 + e_c)(\tilde{w}^{\mathrm{T}} \nabla S \dot{\bar{\xi}} + \frac{1}{2} \parallel \tilde{w} \parallel_{\bar{D}}^2) \\ &+ \alpha_c \nabla J_s^{\mathrm{T}} \dot{\xi} \\ = &-\frac{3}{4} \parallel \tilde{w} \parallel_{\bar{D}}^2 (\tilde{w}^{\mathrm{T}} \nabla S \dot{\bar{\xi}}) - e_c(\tilde{w}^{\mathrm{T}} \nabla S \dot{\bar{\xi}} + \frac{1}{2} \parallel \tilde{w} \parallel_{\bar{D}}^2) \\ &-\left(\frac{1}{8} \parallel \tilde{w} \parallel_{\bar{D}}^4 + (\tilde{w}^{\mathrm{T}} \nabla S \dot{\bar{\xi}})^2\right) + \alpha_c \nabla J_s^{\mathrm{T}} \dot{\xi}\end{aligned} \tag{1G}$$

Assume that $\underline{\lambda}_1 \leq \parallel \nabla S \dot{\bar{\xi}} \parallel \leq \bar{\lambda}_1$, $\parallel e_c \parallel \leq \lambda_2$, $\underline{\lambda}_d \leq \parallel \bar{D} \parallel \leq \bar{\lambda}_d$, $(\tilde{w}^T \nabla S \dot{\bar{\xi}})^2 = \parallel \tilde{w}^T \nabla S \dot{\bar{\xi}} \|^2 \leq \parallel \tilde{w} \|^2 \parallel \nabla S \dot{\bar{\xi}} \|^2 \leq \bar{\lambda}_1^2 \parallel \tilde{w} \|^2$ and $\underline{\lambda}_d^4 \parallel \tilde{w} \|^4 \leq \parallel \tilde{w} \parallel_{\bar{D}}^4 \leq \bar{\lambda}_d^4 \parallel \tilde{w} \|^4$. Using the Young's inequality

$$\pm ab \leq \frac{1}{2\rho^2} a^2 + \frac{\rho^2}{2} b^2 \tag{1H}$$

we have

$$\begin{aligned}&-\frac{3}{4} \parallel \tilde{w} \parallel_{\bar{D}}^2 \left(\tilde{w}^{\mathrm{T}} \nabla S \dot{\bar{\xi}}\right) \leq \frac{3}{8\rho_1^2} \bar{\lambda}_d^4 \parallel \tilde{w} \|^4 + \frac{3}{8} \rho_1^2 \bar{\lambda}_1^2 \parallel \tilde{w} \|^2 \\ &\quad -e_c \left(\tilde{w}^{\mathrm{T}} \nabla S \dot{\bar{\xi}}\right) \leq \frac{1}{2\rho_2^2} \lambda_2^2 + \frac{\rho_2^2}{2} \bar{\lambda}_1^2 \parallel \tilde{w} \|^2 \\ &\quad -e_c \frac{1}{2} \parallel \tilde{w} \parallel_{\bar{D}}^2 \leq \frac{1}{2\rho_3^2} \lambda_2^2 + \frac{\rho_3^2}{2} \bar{\lambda}_d^2 \parallel \tilde{w} \|^4 \\ &-\left(\frac{1}{8} \parallel \tilde{w} \parallel_{\bar{D}}^4 + \left(\tilde{w}^{\mathrm{T}} \nabla S \dot{\bar{\xi}}\right)^2\right) \leq -\left(\frac{1}{8} \underline{\lambda}_d^4 \parallel \tilde{w} \|^4 + \underline{\lambda}_1^2 \parallel \tilde{w} \|^2\right) \\ &\quad \alpha_c \nabla J_s^{\mathrm{T}} \dot{\xi} < -\alpha_c \lambda_J \parallel \nabla J_s \parallel\end{aligned} \tag{1I}$$

Subsequently, the following inequality holds

$$\dot{V} < -\alpha \parallel \tilde{w} \|^4 + \beta \parallel \tilde{w} \|^2 + \gamma - \alpha_c \lambda_J \parallel \nabla J_s \parallel \tag{1J}$$

where $\alpha = \frac{1}{8} \underline{\lambda}_d^4 - \left(\frac{3}{8\rho_1^2} \bar{\lambda}_d^4 + \frac{\rho_3^2}{2} \bar{\lambda}_d^2\right)$, $\beta = \frac{3}{8} \rho_1^2 \bar{\lambda}_1^2 + \frac{\rho_2^2}{2} \bar{\lambda}_1^2 - \underline{\lambda}_1^2$ and $\gamma = \left(\frac{1}{2\rho_2^2} + \frac{1}{2\rho_3^2}\right) \lambda_2^2$. ρ_i are positive numbers required to be chosen appropriately such that $\alpha > 0$

$$\parallel \tilde{w} \parallel \geq \sqrt{\frac{\beta + \sqrt{\beta^2 + 4\alpha\gamma}}{2\alpha}} \tag{1K}$$

$$\parallel \nabla J_s \parallel \geq \frac{\alpha\beta^2 + 4\alpha^2\gamma}{4\alpha^2 \lambda_s \alpha_c} \tag{1L}$$

Finally, if either of the above two inequalities is held, $\dot{V} < 0$.

To conclude, if equation (1K) or (1L) is satisfied when $h = 0$, then in both cases, the time derivative of the Lyapunov candidate in equation (1C) is negative which implies the convergence of $\tilde{w}$ and ξ. This completes the stability analysis.