Reinforcement learning control for coordinated manipulation of multi-robots

Article (Accepted Version)

Reinforcement Learning Control for Coordinated Manipulation of Multi-Robots

Yanan Li[a], Long Chen[b], Keng Peng Tee[a], Qingquan Li[c]

[a]*Institute for Infocomm Research, Agency for Science, Technology and Research, Singapore 138632*
[b]*School of Mobile Information Engineering, Sun Yat-Sen University, Zhuhai 519082, China*
[c]*Shenzhen Key Laboratory of Spatial Smart Sensing and Services, Shenzhen University, Shenzhen 518060, China*

Abstract

In this paper, coordination control is investigated for multi-robots to manipulate an object with a common desired trajectory. Both trajectory tracking and control input minimization are considered for each individual robot manipulator, such that possible disagreement between different manipulators can be handled. Reinforcement learning is employed to cope with the problem of unknown dynamics of both robots and the manipulated object. It is rigorously proven that the proposed method guarantees the coordination control of the multi-robots system under study. The validity of the proposed method is verified through simulation studies.

Keywords: multi-robots coordination, reinforcement learning, robot control

1. Introduction

Coordinated manipulation of multi-robots has attracted researchers' attention as it provides better rigidity and feasibility compared to manipulation of a single robot, yet it brings along challenging control problems [1]. Different from control of a single robot manipulator, a coordination scheme is needed to avoid possible disagreement between multi-robots, which will lead to undesired results, e.g., large internal forces [2]. Typical coordination control schemes include hybrid position/force control and leader-follower control [3]. Hybrid position/force control considers the position of the manipulated object to be in a certain workspace, and the internal force to be within a

small range around the origin. In comparison, the leader-follower method introduces a leader individual, which is followed by other manipulators. Regrading these two coordination control schemes, while the former requires the separation of directions for position and force controls [4], the latter needs multi-robots to communicate with each other through different interfaces. Enlightened by the idea of optimal control, i.e., to achieve the trajectory tracking and simultaneously to penalize the control effort, we propose a coordination scheme in this paper to avoid limitations in existing methods. In particular, when manipulating a common object by multi-robots, each individual aims to track a prescribed trajectory while it complies to others by penalizing its own control effort. This will lead to an optimization-like problem which cannot be handled by conventional optimal control, e.g., linear quadratic regulator (LQR) [5], due to uncertain and nonlinear system dynamics. In the literature, reinforcement learning, also known as adaptive dynamic programming, has been extensively studied in the control community to address this issue [6, 7].

The idea of reinforcement learning is inspired by the phenomena that human beings and other animals always learn from experience through reward and punishment results for survival and growth [8, 9, 10, 11]. In particular, biological experiments show that the dopamine neurotransmitter acts as a reinforcement signal which favors learning at the neuron level [12]. Based on reinforcement learning, a control signal can be generated for an agent to interact with unknown environments. Typically, a cost function or a reward function is defined to describe the control objective, and a control scheme is developed to minimize/maximize the defined cost/reward function [13]. Therefore, a reinforcement learning control can be developed in the form of a composition of two parts: a critic network and an actor network. A critic network is developed to approximate the cost function, while an actor network plays a role to minimize the cost function. Reinforcement learning control has been developed in both continuous-time and discrete-time domains. In [14], a reinforcement learning control has been proposed for systems in continuous time and space. In [15], a state observer is introduced to estimate the future state for the design of adaptive dynamic programming for unknown nonlinear continuous-time systems. In [16], a discrete-time reinforcement learning control is proposed with Lyapunov stability analysis. In [17], optimal control is proposed for unknown nonaffine discrete-time systems by employing adaptive dynamic programming. Reinforcement learning control has also been investigated in control of robots. In [18], a natural actor-critic algorithm is

adopted for the learning of proper impedance for robots in interacting with unknown environments. In [19], the algorithm of policy improvement with path integrals is integrated with reinforcement learning to achieve variable impedance control. In [20], impedance adaptation for robot control is developed based on adaptive dynamic programming proposed in [21]. Literature reviews of reinforcement learning can be found in [22, 23], which introduce the use of reinforcement learning in feedback control and state open challenges of developing a reinforcement learning control.

Based on the above discussions, in this paper, we will introduce a reinforcement learning control for coordinated manipulation of multi-robots. First, a cost function is defined to describe the tracking objective of each individual robot manipulator and its compliance to others. Then, the coordination problem of multi-robots will be transformed to an optimization-like problem. A reinforcement learning control will be designed to minimize the defined cost function, in the presence of unknown system dynamics. Eventually, through Lyapunov stability analysis, the performance of the proposed method will be discussed in details.

The contributions of this paper are highlighted as follows:

(i) the problem of multi-robots coordination is formulated such that both the tracking objective of each individual robot manipulator and its compliance to others are described, with neither the separation of task spaces nor extra communication interfaces;

(ii) system dynamics are transformed to a general model similar to that of a single robot manipulator for the feasibility of control design; and

(iii) a reinforcement learning control is developed subject to unknown dynamics of robot manipulators and object, which guarantee the coordination control of multi-robots.

The rest of the paper is organized as follows. In Section 2, the problem of coordination control under study is formulated. In Section 3, transformation of system dynamics and design of a reinforcement learning control are detailed, followed by the rigorous performance analysis. In Section 4, the validity of the proposed method is verified through simulation studies. Section 5 concludes this paper.

2. Problem Formulation

2.1. System Description

The system under study includes n individual robot manipulators and a rigid object, where the object is tightly grasped by the end-effector of each robot manipulator. It is assumed that there is no relative motion between the robot manipulators and object.

The dynamics of the object in the task space are described as

$$\begin{aligned} m_o\ddot{p} - m_o g &= f_o \\ I_o\dot{\omega} + \omega \times I_o\omega &= \tau_o \end{aligned} \tag{1}$$

where m_o and I_o are the mass and inertia matrix of the manipulated object, p and ω are the position and angular velocity of the object, f_o and τ_o are the force and torque applied to the mass center of the object, and g is the gravitational acceleration.

Define $x_o = [p^T, \theta^T]^T$ where $\dot{\theta} = \omega$, and we have $\dot{x}_o = [\dot{p}^T, \omega^T]^T$. Then, the dynamics of the object can be rewritten in the following form [24]:

$$M_o\ddot{x}_o + C_o(\dot{x}_o)\dot{x}_o + G_o = F_o \tag{2}$$

where $M_o = \begin{bmatrix} m_o I & 0 \\ 0 & I_o \end{bmatrix} \in \mathbb{R}^{m\times m}$, $C_o(\dot{x}_o)\dot{x}_o = \begin{bmatrix} 0 \\ \omega \times I_o\omega \end{bmatrix} \in \mathbb{R}^m$, $G_o = \begin{bmatrix} -m_o g \\ 0 \end{bmatrix} \in \mathbb{R}^m$, and $F_o(t) = \begin{bmatrix} f_o \\ \tau_o \end{bmatrix} \in \mathbb{R}^m$.

Property 1. *The matrix $C_o(\dot{x}_o)$ is skew-symmetric, i.e., $\varrho^T C_o(\dot{x}_o)\varrho = 0$, for $\forall \varrho \in \mathbb{R}^m$.*

The forward kinematics of the i-th robot manipulator is described by $x_i = \varphi_i(q_i)$, where $x_i(t) \in \mathbb{R}^{m_i}$ and $q_i \in \mathbb{R}^{m_i}$ are positions/oritations in the Cartesian space and joint coordinates in the joint space, respectively. Differentiating $x_i = \phi(q_i)$ with respect to time results in $\dot{x}_i = J_{r,i}(q_i)\dot{q}_i$, where $J_{r,i}(q_i) \in \mathbb{R}^{m_i\times m_i}$ is the Jacobian matrix for the i-th robot manipulator. Besides, $J_i(x_o)$ is the Jacobian matrix which describes the kinematic relationship between the mass center of the object and the end-effector of the i-th robot manipulator.

Assumption 1. *The Jacobian matrices $J_{r,i}(q_i)$ and $J_i(x_o)$ are nonsingular in a finite workspace.*

The dynamics of the i-th robot manipulator in the joint space are

$$M_{r,i}(q_i)\ddot{q}_i + C_{r,i}(q_i,\dot{q}_i)\dot{q}_i + G_{r,i}(q_i) + J_{r,i}^T(q_i)F_i = u_{r,i}, \quad i = 1,2,3...,n \tag{3}$$

where $M_{r,i}(q_i) \in \mathbb{R}^{m_i \times m_i}$ is the inertia matrix, $C_{r,i}(q_i,\dot{q}_i)\dot{q}_i \in \mathbb{R}^{m_i}$ denotes the Coriolis and Centrifugal force, $G_{r,i}(q_i) \in \mathbb{R}^{m_i}$ is the gravitational force, F_i denotes the force exerted on the object by the end-effector of the i-th robot manipulator at the interaction point, and $u_{r,i} \in \mathbb{R}^{m_i}$ is the control input.

By considering the Jacobian matrix $J_{r,i}(q_i)$, the dynamics of the i-th robot manipulator can be described in the Cartesian space as below

$$M_i(q_i)\ddot{x}_i + C_i(q_i,\dot{q}_i)\dot{x}_i + G_i(q_i) + F_i = u_i, \quad i = 1,2,3...,n \tag{4}$$

where

$$\begin{aligned} M_i(q_i) &= J_{r,i}^{-T}(q_i)M_{r,i}(q_i)J_{r,i}^{-1}(q_i) \\ C_i(q_i,\dot{q}_i) &= J_{r,i}^{-T}(q_i)(C_{r,i}(q_i,\dot{q}_i) - M_{r,i}(q_i,\dot{q}_i)J_{r,i}^{-1}(q_i)\dot{J}_{r,i}(q_i))J_{r,i}^{-1}(q_i) \\ G_i(q_i) &= J_{r,i}^{-T}(q_i)G_{r,i}(q_i),\ u_i = J_{r,i}^{-T}(q_i)u_{r,i} \end{aligned} \tag{5}$$

Property 2. *[25] The matrix $M_i(q_i)$ is symmetric and positive definite.*

Property 3. *[25] The matrix $\dot{M}_i(q_i) - 2C_i(q_i,\dot{q}_i)$ is skew-symmetric if $C_i(q_i,\dot{q}_i)$ is in Christoffel form, i.e. $\varrho^T(\dot{M}_i(q_i) - 2C_i(q_i,\dot{q}_i))\varrho = 0$, for $\forall \varrho \in \mathbb{R}^{m_i}$.*

The control objective of this work is to let the object move along a desired trajectory x_d while minimizing the control efforts of all robot manipulators. In particular, we define the following cost function

$$\Upsilon(t) = \int_t^\infty c(s)ds \tag{6}$$

where $c(t)$ is an instant cost function defined as

$$c(t) = (x_o - x_d)^T Q_1 (x_o - x_d) + \dot{x}_o^T Q_2 \dot{x}_o + \sum_{i=1}^n u_{r,i}^T R_i u_{r,i} \tag{7}$$

where $Q_1 \geq 0$, $Q_2 \geq 0$, and $R_i > 0$.

Remark 1. *The rule of thumb to choose Q_1 and R_i are as follows: a larger value for Q_1 indicates that a more accurate tracking performance is expected, a larger value for Q_2 indicates that a smoother motion is desirable, and a larger value for R_i indicates that the load of the i-th robot manipulator is expected to be smaller. For example, it is usually to allocate a larger load to a "stronger" robot manipulator (with larger mass and inertia of moment), so R_i should be given a smaller value for this robot manipulator.*

2.2. Preliminary

For the approximation of a continuous function $h(Z) : \mathbb{R}^r \to \mathbb{R}$, the following Radial Function Basis (RBF) Neural Networks (NN) are used [26]:

$$h(Z) = W^{*T} S(Z) + \epsilon \tag{8}$$

where $Z \in \mathbb{R}^r$ is the NN input vector, W^* is the ideal NN weight, and ϵ is the approximation error under the ideal NN weight. $S(Z) = [s_1(Z), ...s_l(Z)]$ is a vector where $s_i(Z)$ is chosen as the Gaussian function for $i = 1, 2, ...l$, which is expressed as

$$s_i(Z) = \exp[\frac{-(Z - \mu_i)^T (Z - \mu_i)}{\eta_i^2}] \tag{9}$$

where μ_i is the designed center for the i-th input element of the NN, and η_i the width of the Gaussian function.

Lemma 1. *[27] Consider a positive function given by*

$$V(t) = \frac{1}{2} e^T(t) \Lambda(t) e(t) + \frac{1}{2} \tilde{W}^T(t) \Gamma^{-1}(t) \tilde{W}(t) \tag{10}$$

where $e(t) = \xi(t) - \xi^*(t)$ and $\tilde{W}(t) = \hat{W}(t) - W^*$, and $\Lambda(t) = \Lambda^T(t) > 0$ and $\Gamma(t) = \Gamma^T(t) > 0$ are dimensionally compatible matrices. If the following inequality holds:

$$\dot{V}(t) \leq \rho V(t) + \kappa \tag{11}$$

where ρ and κ are positive constants, then, given any initial compact set defined by

$$\Omega_0 = \{\xi(0), \xi^*(0), \hat{W}(0) | \xi(0), \hat{W}(0) \text{ finite}, \xi^*(0) \in \Omega^*\} \tag{12}$$

we can conclude that the states and weights will eventually converge to the compact sets defined by

$$\Omega_s = \{\xi(t), \hat{W}(t) | \lim_{t \to \infty} \|e(t)\| = \mu_e^*, \lim_{t \to \infty} \|\tilde{W}\| = \mu_{\tilde{W}}^*\} \tag{13}$$

where constants

$$\begin{aligned} \mu_e^* &= \sqrt{\frac{2\kappa}{\rho \lambda_{\Lambda \min}}} \\ \mu_{\tilde{W}}^* &= \sqrt{\frac{2\kappa}{\rho \lambda_{\Gamma \min}}} \end{aligned} \tag{14}$$

with $\lambda_{\Lambda \min} = \min_{v \in [0,t]} \lambda_{\min}(\Lambda(v))$, and $\lambda_{\Gamma \min} = \min_{v \in [0,t]} \lambda_{\min}(\Gamma^{-1}(v))$.

3. Control Design

3.1. System Transformation

By applying the virtual work principle, we can obtain the relationship between the end-effector forces F_i with the force applied to the mass center of the object F_o. In particular, we have

$$(-F_1)^T \delta x_1 + (-F_2)^T \delta x_2 + ... + (-F_n)^T \delta x_n + F_o^T \delta x_o = 0 \tag{15}$$

Since $\delta x_i = J_i(x_o)\delta x_o$, we obtain

$$F_o = \sum_{i=1}^{n} J_i^T(x_o) F_i \tag{16}$$

Since $\ddot{x}_i = \dot{J}_i(x_o)\dot{x}_o + J_i(x_o)\ddot{x}_o$, the dynamics of the i-th robot can be re-described as

$$M_i(q_i)J_i(x_o)\ddot{x}_o + \big(M_i(q_i)\dot{J}_i(x_o) + C_i(q_i,\dot{q}_i)J_i(x_o)\big)\dot{x}_o + G_i(q_i) + F_i = u_i \tag{17}$$

Multiplying both sides of the above equation by $J_i^T(x_o)$, we have

$$\begin{aligned} &J_i^T(x_o)M_i(q_i)J_i(x_o)\ddot{x}_o + J_i^T(x_o)\big(M_i(q_i)\dot{J}_i(x_o) + C_i(q_i,\dot{q}_i)J_i(x_o)\big)\dot{x}_o \\ &+J_i^T(x_o)G_i(q_i) + J_i^T(x_o)F_i = J_i^T(x_o)u_i \end{aligned} \tag{18}$$

By adding the above dynamics of all the robot manipulators together, we obtain the following combined dynamics

$$\begin{aligned} &\sum_{i=1}^{n} J_i^T(x_o)M_i(q_i)J_i(x_o)\ddot{x}_o + \sum_{i=1}^{n} J_i^T(x_o)\big(M_i(q_i)\dot{J}_i(x_o) + C_i(q_i,\dot{q}_i)J_i(x_o)\big)\dot{x}_o \\ &+\sum_{i=1}^{n} J_i^T(x_o)G_i(q_i) + \sum_{i=1}^{n} J_i^T(x_o)F_i = \sum_{i=1}^{n} J_i^T(x_o)u_i \end{aligned} \tag{19}$$

Considering Eq. (16) and substituting the dynamics of the object, i.e., Eq. (2), into Eq. (19), we obtain the model of the overall system as below

$$\begin{aligned} &\big(\sum_{i=1}^{n} J_i^T(x_o)M_i(q_i)J_i(x_o) + M_o\big)\ddot{x}_o + \big(\sum_{i=1}^{n} J_i^T(x_o)\big(M_i(q_i)\dot{J}_i(x_o) \\ &+C_i(q_i,\dot{q}_i)J_i(x_o)\big) + C_o(\dot{x}_o)\big)\dot{x}_o + \big(\sum_{i=1}^{n} J_i^T(x_o)G_i(q_i) + G_o\big) \\ &= \sum_{i=1}^{n} J_i^T(x_o)u_i \end{aligned} \tag{20}$$

By defining

$$\begin{aligned} M(q_i, x_o) &= \sum_{i=1}^{n} J_i^T(x_o) M_i(q_i) J_i(x_o) + M_o \\ C(q_i, \dot{q}_i, x_o, \dot{x}_o) &= \sum_{i=1}^{n} J_i^T(x_o) \big(M_i(q_i) \dot{J}_i(x_o) + C_i(q_i, \dot{q}_i) J_i(x_o) \big) + C_o(\dot{x}_o) \\ G(q_i, x_o) &= \sum_{i=1}^{n} J_i^T(x_o) G_i(q_i) + G_o \\ u &= \sum_{i=1}^{n} J_i^T(x_o) u_i \end{aligned} \tag{21}$$

the above system model can be rewritten as

$$M(q_i, x_o)\ddot{x}_o + C(q_i, \dot{q}_i, x_o, \dot{x}_o)\dot{x}_o + G(q_i, x_o, \dot{x}_o) = u \tag{22}$$

According to Properties 1, 2 and 3, we immediately obtain the following properties for the combined system (22).

Property 4. *The matrix $M(q_i, x_o)$ is symmetric and positive definite.*

Property 5. *The matrix $\dot{M}(q_i, x_o) - 2C(q_i, \dot{q}_i, x_o, \dot{x}_o)$ is skew-symmetric, i.e. $\varrho^T(\dot{M}(q_i, x_o) - 2C(q_i, \dot{q}_i, x_o, \dot{x}_o))\varrho = 0$, for $\forall \varrho \in \mathbb{R}^m$.*

Proof 1.

$$\begin{aligned} & \dot{M}(q_i, x_o) - 2C(q_i, \dot{q}_i, x_o, \dot{x}_o) \\ = & \sum_{i=1}^{n} (2J_i^T(x_o) M_i(q_i) \dot{J}_i(x_o) + J_i^T(x_o) \dot{M}_i(q_i) J_i(x_o)) + \dot{M}_o \\ & - \sum_{i=1}^{n} 2J_i^T(x_o)(M_i(q_i)\dot{J}_i(x_o) + C_i(q_i, \dot{q}_i) J_i(x_o)) - 2C_o(\dot{x}_o) \\ = & \sum_{i=1}^{n} J_i^T(x_o)(\dot{M}_i(q_i) - 2C_i(q_i, \dot{q}_i)) J_i(x_o) - 2C_o(\dot{x}_o) \end{aligned} \tag{23}$$

Recalling Properties 3 and 1 of robot manipulators and object dynamics, we can obtain

$$\begin{aligned} & x_o^T(\dot{M}(q_i,x_o)-2C(q_i,\dot{q}_i,x_o,\dot{x}_o))x_o \\ = & \sum_{i=1}^{n}(J_i(x_o)x_o)^T(\dot{M}_i(q_i)-2C_i(q_i,\dot{q}_i))J_i(x_o)x_o + x_o^T(\dot{M}_o-2C_o(\dot{x}_o))x_o \\ = & \sum_{i=1}^{n} x_i^T(\dot{M}_i(q_i)-2C_i(q_i,\dot{q}_i))x_i - 2x_o^T C_o(\dot{x}_o)x_o = 0 \end{aligned} \tag{24}$$

for $\forall x_o \in \mathbb{R}^m$. Replacing x_o by ϱ completes the proof.

Similarly as in [20], we consider the desired trajectory generated by the following system:

$$\begin{cases} \dot{z} = Uz \\ x_d = Yz \end{cases}$$

where $z \in R^m$ is an auxiliary variable, $U \in R^{m\times m}$ and $Y \in R^{m\times m}$ are known matrices, and (U, Y) is observable. Then, the instant cost function defined in (7) becomes

$$\begin{aligned} c(t) &= \begin{bmatrix} x_o^T & x_d^T \end{bmatrix} \begin{bmatrix} Q_1 & -Q_1 \\ -Q_1 & Q_1 \end{bmatrix} \begin{bmatrix} x_o \\ x_d \end{bmatrix} + \dot{x}_o^T Q_2 \dot{x}_o + \sum_{i=1}^{n} u_{r,i}^T R_i u_{r,i} \\ &= \begin{bmatrix} x_o^T & z^T \end{bmatrix} \begin{bmatrix} Q_1 & -Q_1 Y \\ -Y^T Q_1 & Y^T Q_1 Y \end{bmatrix} \begin{bmatrix} x_o \\ z \end{bmatrix} + \dot{x}_o^T Q_2 \dot{x}_o \\ &\quad + \sum_{i=1}^{n} u_{r,i}^T R_i u_{r,i} \end{aligned} \tag{25}$$

Denote $\xi = [x_o^T, z^T, \dot{x}_o^T]^T$, $u' = [u_1^T, u_2^T, ...u_n^T]^T$, $R' = \mathrm{diag}[(J_{r,i}(q_i)R_i J_{r,i}^T(q_i))]$ for $i = 1, 2, ...n$, and

$$Q = \begin{bmatrix} Q_1 & -Q_1 Y & 0 \\ -Y^T Q_1 & Y^T Q_1 Y & 0 \\ 0 & 0 & Q_2 \end{bmatrix} \tag{26}$$

Then, we obtain

$$c(t) = \xi^T Q \xi + u'^T R' u' \tag{27}$$

Denote $J' = [J_1^T, J_2^T, ...J_n^T]^T$. From the definition of u in (21), we know that $u = J'^T u'$. Therefore, the above instant cost function is finally written as

$$c(t) = \xi^T Q \xi + u^T R u \tag{28}$$

where $R = J'^{\dagger} R' J'^{\dagger T}$ with $J'^{\dagger}$ as the pseudoinverse of J'.

Following the transformation in this subsection, the coupled dynamics of multi robot manipulators and object are described in a unified form, i.e., Eq. (22). Then, it becomes straightforward to design a control for such a general system to minimize the cost function (6). Since $M(q_i, x_o)$, $C(q_i, \dot{q}_i, x_o, \dot{x}_o)$, and $G(q_i, x_o)$ are unknown and typically nonlinear due to the involvement of robot and object dynamics, we develop reinforcement learning for the control design of system (22), as detailed in the following subsection.

3.2. Reinforcement Learning

First, a critic network is used to approximate the cost function at current state, i.e.,

$$\begin{aligned} \Upsilon(t) &= W_c^{*T} S_c(Z_c) + \epsilon_c \\ \hat{\Upsilon}(t) &= \hat{W}_c^T S_c(Z_c) \end{aligned} \tag{29}$$

where $Z_c = \xi$, and other denotations follow NN denotations in Section 2.2. An ideal approximation is achieved if the following error is eliminated:

$$E_c \triangleq \frac{1}{2}(c - \hat{W}_c^T \dot{S}_c)^2 \tag{30}$$

Therefore, the updating law for the critic network is designed using the gradient descent method, as below

$$\begin{aligned} \dot{\hat{W}}_c &= -\sigma_c \frac{\partial E_c}{\partial \hat{W}_c} \\ &= \sigma_c (c - \hat{W}_c^T \dot{S}_c) \dot{S}_c \end{aligned} \tag{31}$$

where $\sigma_c > 0$ is the learning rate for the critic network.

Second, we introduce an action network to achieve the control objective discussed in Section 2.1. Define $e = x_o - x_d$ as the tracking error, and the corresponding Lyapunov function candidate is $V_1 = \frac{1}{2} e^T e$. Its time derivative is

$$\begin{aligned} \dot{V}_1 &= e^T \dot{e} \\ &= e^T (\dot{x}_o - \dot{x}_d - K_1 e + K_1 e) \end{aligned} \tag{32}$$

where K_1 is a positive definite matrix. By defining $\dot{x}_r = \dot{x}_d - K_1 e$ and $e_v = \dot{x}_o - \dot{x}_r$, we have

$$\dot{V}_1 = -e^T K_1 e + e^T e_v \tag{33}$$

Considering the system dynamics (22) and another Lyapunov function candidate $V_2 = \frac{1}{2} e_v^T M e_v$, we have

$$\begin{aligned} \dot{V}_2 &= \frac{1}{2} e_v^T \dot{M} e_v + e_v^T M \dot{e}_v \\ &= \frac{1}{2} e_v^T \dot{M} e_v + e_v^T (-C\dot{x}_o - G + u - M\ddot{x}_r) \\ &= e_v^T (-M\ddot{x}_r - C\dot{x}_r - G + u) \end{aligned} \tag{34}$$

where Property 5 is applied. It is trivial to design an ideal control $u^* = M\ddot{x}_r + C\dot{x}_r + G - e - K_2 e_v$, where K_2 is a positive definite matrix. However, since M, C and G are unknown, an action network is introduced to approximate the unknown parts of control input, as follows

$$M\ddot{x}_r + C\dot{x}_r + G = W_a^{*T} S_a(Z_a) + \epsilon_a \tag{35}$$

where $Z_a = [q_i, \dot{q}_i, x_o, \dot{x}_o, \dot{x}_r, \ddot{x}_r]$. Then, the ideal control input becomes

$$u^* = W_a^{*T} S_a(Z_a) + \epsilon_a - e - K_2 e_v \tag{36}$$

and the actual control is designed as

$$u = \hat{W}_a^T S_a(Z_a) - e - K_2 e_v \tag{37}$$

Considering a Lyapunov function candidate for both e and e_v, i.e. $V = V_1 + V_2$, we have

$$\dot{V} = \dot{V}_1 + \dot{V}_2 = -e^T K_1 e - e_v^T K_2 e_v + e_v^T (\tilde{W}_a^T S_a(Z_a) - \epsilon_a) \tag{38}$$

where $\tilde{W}_a = \hat{W}_a - W_a^*$ is the approximation error of NN weights. Since the objective of updating law of $\tilde{W}_a$ is to minimize the approximation error $\tilde{W}_a$ itself and the estimated cost function $\hat{\Upsilon}$, we define an error variable $e_a = \sum_{i=1}^{m} \tilde{W}_{a,i}^T S_a + k_\Upsilon \hat{\Upsilon}$, where $\tilde{W}_{a,i}$ is the i-th column of $\tilde{W}_a$ and k_Υ is a positive scalar. Again, using the gradient descent method, we obtain the updating law of action network as below

$$\dot{\hat{W}}_{a,i} = -\sigma_a (\hat{W}_{a,i}^T S_a + k_\Upsilon \hat{\Upsilon}) S_a \tag{39}$$

where σ_a is a designed learning rate for the action network.

3.3. Stability Analysis

Consider a Lyapunov function candidate as below

$$V = V_1 + V_2 + V_c + V_a \tag{40}$$

where $V_c = \frac{1}{2}\tilde{W}_c^T\tilde{W}_c$ and $V_a = \frac{1}{2}e_a^T e_a$. Its time derivative is

$$\begin{aligned}
\dot{V} &= \dot{V}_1 + \dot{V}_2 + \dot{V}_c + \dot{V}_a \\
&= -e^T K_1 e - e_v^T K_2 e_v + e_v^T(\tilde{W}_a^T S_a - \epsilon_a) + \tilde{W}_c^T\dot{\tilde{W}}_c + e_a^T\dot{e}_a \\
&= -e^T K_1 e - e_v^T K_2 e_v + e_v^T(\tilde{W}_a^T S_a - \epsilon_a) \\
&\quad +\tilde{W}_c^T\dot{\tilde{W}}_c + e_a^T(\sum_{i=1}^{m}\dot{\hat{W}}_{a,i}^T\frac{\partial e_a}{\partial\hat{W}_{a,i}}) \\
&= -e^T K_1 e - e_v^T K_2 e_v + e_v^T(\tilde{W}_a^T S_a - \epsilon_a) \\
&\quad +\tilde{W}_c^T\dot{\tilde{W}}_c + e_a(\sum_{i=1}^{m}\dot{\hat{W}}_{a,i}^T)S_a
\end{aligned} \tag{41}$$

Considering the updating law for the action network, i.e., Eq. (39), we have

$$\begin{aligned}
\dot{V} &= -e^T K_1 e - e_v^T K_2 e_v + e_v^T(\tilde{W}_a^T S_a - \epsilon_a) + \tilde{W}_c^T\dot{\tilde{W}}_c \\
&\quad -\sigma_a S_a^T S_a e_a\sum_{i=1}^{m}(\hat{W}_{a,i}^T S_a + k_\Upsilon\hat{\Upsilon}) \\
&= -e^T K_1 e - e_v^T K_2 e_v + e_v^T\tilde{W}_a^T S_a - e_v^T\epsilon_a + \tilde{W}_c^T\dot{\hat{W}}_c \\
&\quad -\sigma_a S_a^T S_a e_a\sum_{i=1}^{m}(W_{a,i}^{*T} S_a + e_a) \\
&= -e^T K_1 e - e_v^T K_2 e_v + e_v^T\tilde{W}_a^T S_a - e_v^T\epsilon_a + \tilde{W}_c^T\dot{\hat{W}}_c - \sigma_a S_a^T S_a e_a^2 \\
&\quad -\sigma_a S_a^T S_a e_a\sum_{i=1}^{m}W_{a,i}^{*T} S_a
\end{aligned} \tag{42}$$

Considering the updating law for the critic network, i.e., Eq. (31), we have

$$\begin{aligned}
\dot{V} &= -e^T K_1 e - e_v^T K_2 e_v + e_v^T \tilde{W}_a^T S_a - e_v^T \epsilon_a + \sigma_c \tilde{W}_c^T (c - \hat{W}_c^T \dot{S}_c)\dot{S}_c \\
&\quad -\sigma_a S_a^T S_a e_a^2 - \sigma_a S_a^T S_a e_a \sum_{i=1}^{m} W_{a,i}^{*T} S_a \\
&= -e^T K_1 e - e_v^T K_2 e_v + e_v^T \tilde{W}_a^T S_a - e_v^T \epsilon_a + \sigma_c \tilde{W}_c^T (W_c^{*T} \dot{S}_c + \dot{\epsilon}_c - \hat{W}_c^T \dot{S}_c)\dot{S}_c \\
&\quad -\sigma_a S_a^T S_a e_a^2 - \sigma_a S_a^T S_a e_a \sum_{i=1}^{m} W_{a,i}^{*T} S_a \\
&= -e^T K_1 e - e_v^T K_2 e_v + e_v^T \tilde{W}_a^T S_a - e_v^T \epsilon_a - \sigma_c \tilde{W}_c^T (\tilde{W}_c^T \dot{S}_c + \dot{\epsilon}_c)\dot{S}_c \\
&\quad -\sigma_a S_a^T S_a e_a^2 - \sigma_a S_a^T S_a e_a \sum_{i=1}^{m} W_{a,i}^{*T} S_a \\
&\leq -e^T K_1 e - e_v^T K_2 e_v + e_v^T \tilde{W}_a^T S_a - e_v^T \epsilon_a - \frac{1}{2}\sigma_c \dot{S}_c^T \dot{S}_c \tilde{W}_c^T \tilde{W}_c + \frac{\sigma_c \dot{\epsilon}_c^2}{2} \\
&\quad -\sigma_a S_a^T S_a e_a^2 - \sigma_a S_a^T S_a e_a \sum_{i=1}^{m} W_{a,i}^{*T} S_a \qquad (43)
\end{aligned}$$

Substituting inequalities

$$\begin{aligned}
-e_v^T \epsilon_a &\leq \frac{\|e_v\|^2}{2} + \frac{\epsilon_a^2}{2} \\
-S_a^T S_a e_a \sum_{i=1}^{m} W_{a,i}^{*T} S_a &\leq \frac{S_a^T S_a e_a^2}{2} + \frac{\| \sum_{i=1}^{m} W_{a,i}^{*T} S_a \|^2}{2} \leq \frac{e_a^2}{2} + \|W_a^*\|^2 \\
e_v^T \tilde{W}_a^T S_a &\leq \frac{\|e_v\|^2}{2} + \frac{\|\tilde{W}_a^T S_a\|^2}{2} \\
&\leq \frac{\|e_v\|^2}{2} + e_a^2 + k_\Upsilon^2 \hat{\Upsilon}^2 \\
&\leq \frac{\|e_v\|^2}{2} + e_a^2 + \frac{k_\Upsilon^2 \|W_c^*\|^2}{2} + \frac{k_\Upsilon^2 \|\tilde{W}_c\|^2}{2} \qquad (44)
\end{aligned}$$

to the above inequality leads to

$$\begin{aligned}
\dot{V} &\leq -e_1^T K_1 e - e_v^T K_2 e_v + \frac{\|e_v\|^2}{2} + e_a^2 + \frac{k_\Upsilon^2 \|W_c^*\|^2}{2} + \frac{k_\Upsilon^2 \|\tilde{W}_c\|^2}{2} \\
&\quad + \frac{\|e_v\|^2}{2} + \frac{\epsilon_a^2}{2} - \frac{1}{2}\sigma_c \dot{S}_c^T \dot{S}_c \tilde{W}_c^T \tilde{W}_c + \frac{\sigma_c \dot{\epsilon}_c^2}{2} \\
&\quad - \sigma_a S_a^T S_a e_a^2 + \frac{\sigma_a e_a^2}{2} + \sigma_a \|W_a^*\|^2 \\
&\leq -e_1^T K_1 e - e_v^T (K_2 - I) e_v - \frac{\sigma_c \dot{S}_c^T \dot{S}_c - k_\Upsilon^2}{2} \tilde{W}_c^T \tilde{W}_c \\
&\quad - (\sigma_a S_a^T S_a - \frac{\sigma_a + 2}{2}) e_a^2 + \frac{k_\Upsilon^2 \|W_c^*\|^2}{2} \\
&\quad + \frac{\epsilon_a^2}{2} + \frac{\sigma_c \dot{\epsilon}_c^2}{2} + \sigma_a \|W_a^*\|^2
\end{aligned} \tag{45}$$

In accordance with the definition of $V = V_1 + V_2 + V_c + V_a$ with $V_1 = \frac{1}{2} e^T e$, $V_2 = \frac{1}{2} e_v^T M e_v$, $V_c = \frac{1}{2} \tilde{W}_c^T \tilde{W}_c$ and $V_a = \frac{1}{2} e_a^T e_a$, we have

$$\dot{V} \leq -\rho V + \kappa \tag{46}$$

where

$$\begin{aligned}
\rho &= \min\{2K_1, 2\lambda_{M\min}(K_2 - I), \sigma_c \beta_{S_c} - k_\Upsilon^2, 2(\sigma_a \beta_{S_a} - \frac{\sigma_a + 2}{2})\} \\
\kappa &= \frac{k_\Upsilon^2 \beta_c^2}{2} + \frac{\varepsilon_a^2}{2} + \frac{\sigma_c \varepsilon_c^2}{2} + \sigma_a \beta_a^2
\end{aligned} \tag{47}$$

where ε_a, ε_c, β_c, and β_a denote the upper bounds of ϵ_a, $\dot{\epsilon}_c$, W_c^*, and W_a^*, respectively. In addition, $\beta_{S_c} \leq \dot{S}_c^T \dot{S}_c$ and $\beta_{S_a} \leq S_a^T S_a$, which are assured by introducing a noise into the control input such that the persistence excitation condition is satisfied [28].

According to Lemma 1, if the following conditions are satisfied

$$\begin{aligned}
&K_2 - I > 0 \\
&\sigma_c \beta_{S_c} - k_\Upsilon^2 > 0 \\
&\sigma_a \beta_{S_a} - \frac{\sigma_a + 2}{2} > 0
\end{aligned} \tag{48}$$

then, all the closed-loop signals, including e, e_v, $\tilde{W}_c$ and e_a, will remain semi-globally uniformly ultimately bounded. For completeness, multiplying

$\dot{V} = -\rho V + \kappa$ by $e^{\rho t}$, we can obtain

$$\frac{d}{dt}(e^{\rho t}V) \leq \kappa e^{\rho t} \tag{49}$$

By integrating it, we can obtain

$$V \leq (V(0) - \frac{\kappa}{\rho})\kappa e^{-\rho t} + \frac{\kappa}{\rho} \leq V(0) + \frac{\kappa}{\rho} \tag{50}$$

Therefore, signals e, e_v, $\tilde{W}_c$ and e_a remain in the compact set $\Omega_1 := \{\chi | \ \|\chi\| \leq \mu_r\}$ and finally they will converge to the convergence compact set $\Omega_2 := \{\chi | \ \|\chi\| \leq \mu_c\}$, where

$$\mu_r = \sqrt{2(V(0) + \frac{\kappa}{\rho})}, \quad \mu_c = \sqrt{\frac{2\kappa}{\rho}} \tag{51}$$

From definitions of ρ and κ in Eq. (47), we find that sizes of Ω_1 and Ω_2 can be adjusted by choosing different values of design parameters, e.g., K_1, K_2, σ_a, and σ_c. They can be made very small but other effects of improper selection of design parameters should also be considered.

4. Simulation Study

In this section, simulation study is conducted to verify the validity of the proposed method. In particular, two 2-degrees-of-freedom (2-DoF) planar manipulators coordinate to move an object together along a desired trajectory. These two manipulators have same parameters, which are given in Table 1, where m_i, l_i, and I_{zi}, $i = 1, 2$, represent mass, length, and moment of inertia about the axis that comes out of the page passing through the center of mass, respectively. The object under consideration is a square with length $l = 0.1$m, mass $m_o = 0.1$kg, and moment of inertia $I_o = 0.1\text{kgm}^2$. The desired trajectory of the mass center of the object is $x_d = [0.1\cos(t) \ 0.1\sin(t) \ 0]^T$, which indicates that no rotation is expected while the translation motion is a circle. It is generated by Eq. (25) with

$$U = \begin{bmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \ Y = 0.1I \tag{52}$$

The initial position and velocity of the object are $x_o(0) = [0 \ 0 \ 0]^T$ and $\dot{x}_o(0) = [0 \ 0 \ 0]^T$.

Table 1: Parameters of Each Robot Manipulator

Parameter	Description	Value
m_1	Mass of link 1	2.00kg
m_2	Mass of link 2	0.85kg
l_1	Length of link 1	0.30m
l_2	Length of link 2	0.30m
I_{z1}	Moment of inertia of link 1	0.05kgm^2
I_{z2}	Moment of inertia of link 2	0.02kgm^2

Denote joint angles of the first and second robot manipulators as q_1 and q_2, and q_3 and q_4, respectively. Correspondingly, positions of end-effectors of the first and second robot manipulators are x_1 and x_2, and x_3 and x_4, respectively. Then, we have the kinematic relationship from the joint space of each robot manipulator to its corresponding Cartesian space, as below:

$$\begin{aligned} x_1 &= d_1 + l_1 \cos q_1 + l_2 \cos(q_1 + q_2) \\ x_2 &= l_1 \sin q_1 + l_2 \sin(q_1 + q_2) \\ x_3 &= d_2 + l_1 \cos q_3 + l_2 \cos(q_3 + q_4) \\ x_4 &= l_1 \sin q_3 + l_2 \sin(q_3 + q_4) \end{aligned} \tag{53}$$

where $[d_1, 0]^T$ and $[d_2, 0]^T$ are positions of bases of two robot manipulators. For the kinematic relationship from the end-effector of each robot manipulator to the mass center of the object, we have

$$\begin{aligned} x_{o,1} &= \frac{l}{2} \cos\theta + x_1 \\ x_{o,2} &= \frac{l}{2} \sin\theta + x_2 \\ x_{o,1} &= -\frac{l}{2} \cos\theta + x_3 \\ x_{o,2} &= -\frac{l}{2} \sin\theta + x_4 \end{aligned} \tag{54}$$

Therefore, the Jacobian matrices from the joint space of each robot manip-

ulator to its corresponding Cartesian space are

$$J_{r,1}(q) = \begin{bmatrix} -l_1 \sin q_1 - l_2 \sin(q_1 + q_2) & -l_2 \sin(q_1 + q_2) \\ l_1 \cos q_1 + l_2 \cos(q_1 + q_2) & l_2 \cos(q_1 + q_2) \end{bmatrix}$$
$$J_{r,2}(q) = \begin{bmatrix} -l_1 \sin q_3 - l_2 \sin(q_3 + q_4) & -l_2 \sin(q_3 + q_4) \\ l_1 \cos q_3 + l_2 \cos(q_3 + q_4) & l_2 \cos(q_3 + q_4) \end{bmatrix} \quad (55)$$

The Jacobian matrices from the end-effector of each robot manipulator to the mass center of the object are

$$J_1(x_o) = \begin{bmatrix} 1 & 0 & \frac{l}{2} \sin \theta \\ 0 & 1 & -\frac{l}{2} \cos \theta \end{bmatrix}$$
$$J_2(x_o) = \begin{bmatrix} 1 & 0 & -\frac{l}{2} \sin \theta \\ 0 & 1 & \frac{l}{2} \cos \theta \end{bmatrix} \quad (56)$$

For detailed dynamic models of robot manipulators, readers may refer to [25]. For the critic network, centers are evenly distributed in $[-1, 1]$, the variance is 50, and the corresponding number of NN nodes is 2^{10}. For the actor network, centers are also evenly distributed in $[-1, 1]$, the variance is 100, and the number of NN nodes is 2^{16}. The initial NN weights are all set as zeros. Other parameters are set as: $\sigma_a = \sigma_c = 0.1$, $K_1 = 10I$, $K_2 = 5I$, and $k_\Upsilon = 0.2$. A white Gaussian noise of power 0dBW is added into the control input.

Simulation results are shown in Figs. 1 to 4. Fig. 1 demonstrates that the trajectory of the mass center of the object tracks the desired trajectory. Fig. 2 illustrates that the control input is guaranteed to be bounded and it becomes very small when the trajectory tracking in Fig. 1 is achieved. As discussed in the Introduction, since the control input is penalized in the proposed method, it leads to compliance of each robot manipulator with others when there exists disagreement. Besides, norms of estimates of NN weights for critic and actor networks are shown in Fig. 3, which are found to be bounded and eventually converge to certain constants. Correspondingly, from Fig. 4, it is found that the instant cost function defined in Eq. (7) reduces significantly with respect to time. These results well demonstrate that the proposed control achieves the tracking performance of the object manipulated by multi-robots, while the compliance of each individual robot manipulator is also guaranteed.

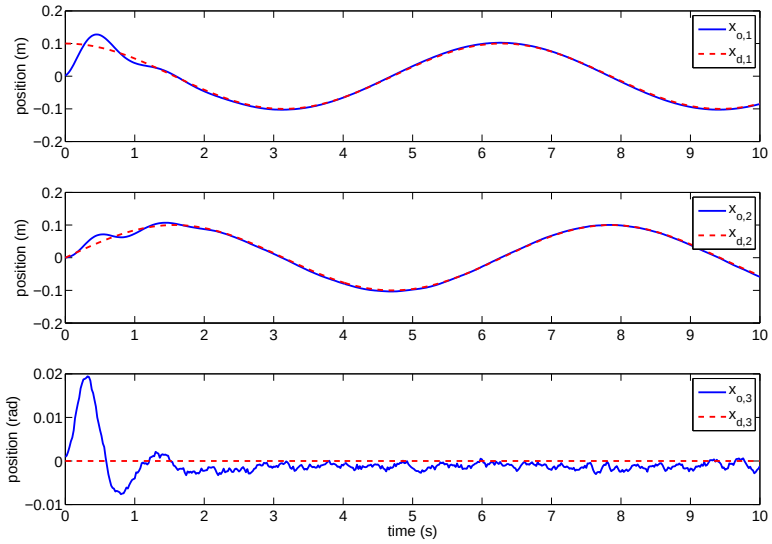

Figure 1: Trajectories of the mass center of the object (top and middle: translation; bottom: rotation)

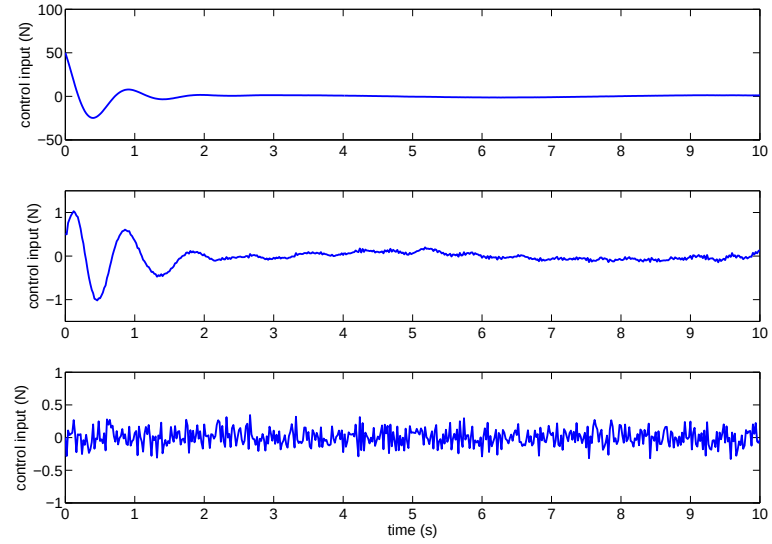

Figure 2: Three components of control input

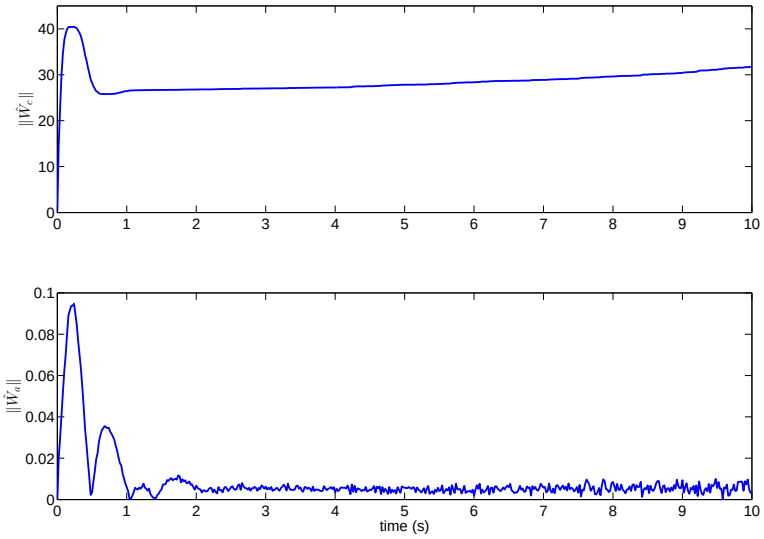

Figure 3: Norms of estimates of NN weights

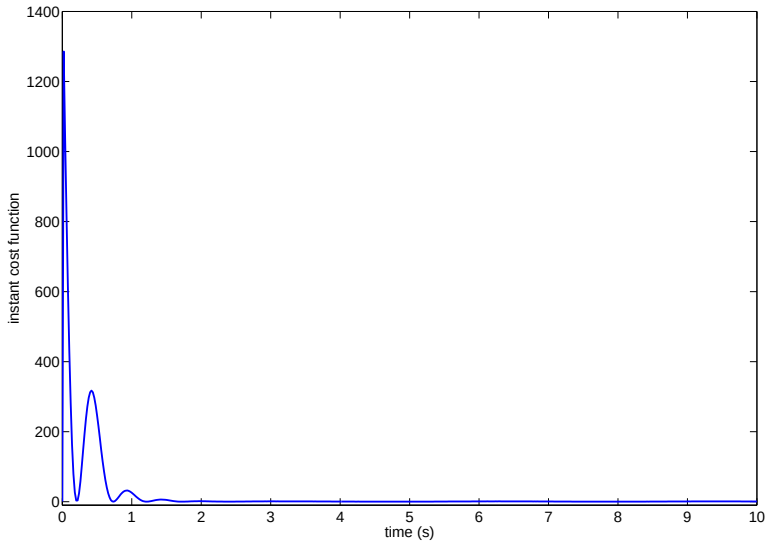

Figure 4: Instant cost function

5. Conclusion

In this paper, we have investigated the control problem for multi-robots coordinated manipulation. By considering both the dynamics of the object and robot manipulators, we have obtained a combined system model which is feasible for the control design. To address the issue of unknown dynamics, we have employed reinforcement learning and developed two neural networks for the control design. Lyapunov's direct method has been used for the performance analysis of the closed-loop system under the proposed control. Two-robots co-manipulation has been considered in simulation to verify the effectiveness of the proposed control in that trajectory tracking of the object is achieved and the control effort of each manipulator is minimized.

6. Acknowledgement

This work is supported by Grant No. 1225100001 from the Science and Engineering Research Council (SERC), A*STAR, Singapore.

[1] J. P. Merlet (Ed.), Parallel robots, Vol. 74, Springer, 2001.

[2] R. Bonitz, T. Hsia, Robust internal-force based impedance control for coordinating manipulators-theory and experiments, in: Proceedings of IEEE International Conference on Robotics and Automation, Vol. 1, 1996, pp. 622–628.

[3] Z. Li, P. Y. Tao, S. S. Ge, M. Adams, W. Wijesoma, Robust adaptive control of cooperating mobile manipulators with relative motion, IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics 39 (1) (2009) 103–116.

[4] Y. Li, S. S. Ge, Impedance learning for robots interacting with unknown environments, IEEE Transactions on Control Systems Technology 22 (4) (2014) 1422–1432.

[5] H. Kwakernaak, R. Sivan, Linear Optimal Control Systems, John Wiley & Sons, Inc. New York, NY, USA, 1972.

[6] R. S. Sutton, A. G. Barto, R. J. Williams, Reinforcement learning is direct adaptive optimal control, IEEE Control Systems 12 (2) (1992) 19–22.

[7] X. Liu, S. Balakrishnan, Convergence analysis of adaptive critic based optimal control, in: Proceedings of American Control Conference, Vol. 3, 2000, pp. 1929–1933.

[8] J. J. Murray, C. J. Cox, G. G. Lendaris, R. Saeks, Adaptive dynamic programming, IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews 32 (2) (2002) 140–153.

[9] M. T. Rosenstein, A. G. Barto, Supervised actor-critic reinforcement learning, Learning and Approximate Dynamic Programming: Scaling Up to the Real World (2004) 359–380.

[10] P. J. Werbos, Using ADP to understand and replicate brain intelligence: The next level design, Proceedings of the 2007 IEEE Symposium on Approximate Dynamic Programming and Reinforcement Learning (2007) 209–216.

[11] P. J. Werbos, Intelligence in the brain: A theory of how it works and how to build it, Neural Networks 22 (3) (2009) 200–212.

[12] D. Vrabie, F. Lewis, Neural network approach to continuous-time direct adaptive optimal control for partially unknown nonlinear systems, Neural Networks 22 (3) (2009) 237–246.

[13] D. P. Bertsekas, D. P. Bertsekas, D. P. Bertsekas, D. P. Bertsekas, Dynamic programming and optimal control, Vol. 1, Athena Scientific Belmont, MA, 1995.

[14] K. Doya, Reinforcement learning in continuous time and space, Neural computation 12 (1) (2000) 219–245.

[15] D. Liu, Y. Huang, D. Wang, Q. Wei, Neural-network-observer-based optimal control for unknown nonlinear systems using adaptive dynamic programming, International Journal of Control 86 (9) (2013) 1554–1566.

[16] P. He, S. Jagannathan, Reinforcement learning neural-network-based controller for nonlinear discrete-time systems with input constraints, IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics 37 (2) (2007) 425–436.

[17] X. Zhang, H. Zhang, Q. Sun, Y. Luo, Adaptive dynamic programming-based optimal control of unknown nonaffine nonlinear discrete-time systems with proof of convergence, Neurocomputing 91 (2012) 48–55.

[18] B. Kim, J. Park, S. Park, S. Kang, Impedance learning for robotic contact tasks using natural actor-critic algorithm, IEEE Transactions on Systems, Man and Cybernetics-Part B: Cybernetics 40 (2) (2010) 433–443.

[19] J. Buchli, F. Stulp, E. Theodorou, S. Schaal, Learning variable impedance control, International Journal of Robotics Research 30 (2011) 820–833.

[20] S. S. Ge, Y. Li, C. Wang, Impedance adaptation for optimal robot-environment interaction, International Journal of Control 87 (2) (2014) 249–263.

[21] Y. Jiang, Z. P. Jiang, Computational adaptive optimal control for continuous-time linear systems with completely unknown dynamics, Automatica 48 (2012) 2699–2704.

[22] F. Lewis, D. Vrabie, Reinforcement learning and adaptive dynamic programming for feedback control, IEEE Circuits and Systems Magazine 9 (3) (2009) 32–50.

[23] Z. P. Jiang, Y. Jiang, Robust adaptive dynamic programming for linear and nonlinear systems: An overview, European Journal of Control 19 (5) (2013) 417–425.

[24] I. D. Walker, R. A. Freeman, S. I. Marcus, Analysis of motion and internal loading of objects grasped by multiple cooperating manipulators, International Journal of Robotics Research 10 (4) (1991) 396–409.

[25] S. S. Ge, T. H. Lee, C. J. Harris, Adaptive Neural Network Control of Robotic Manipulators, World Scientific, London, 1998.

[26] R. M. Sanner, J. E. Slotine, Gaussian networks for direct adaptive control, IEEE Transactions on Neural Networks 3 (6) (1992) 837–863.

[27] S. S. Ge, C. Wang, Adaptive Neural Control of Uncertain MIMO Nonlinear Systems, IEEE Transactions on Neural Networks 15 (3) (2004) 674–692.

[28] M. Green, J. B. Moore, Persistence of excitation in linear systems, Systems & Control Letters 7 (5) (1986) 351–360.