Neurocomputing 170 (2015) 168–175

Contents lists available at ScienceDirect

Neurocomputing

journal homepage: www.elsevier.com/locate/neucom

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, Singapore
[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

ARTICLE INFO

Article history:
Received 17 June 2014
Received in revised form
18 October 2014
Accepted 5 February 2015
Available online 10 July 2015

Keywords:
Multi-robots coordination
Reinforcement learning
Robot control

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.

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 [8–11]. 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

* Corresponding author. Tel.: +86 756 3668787.
E-mail address: chenl46@mail.sysu.edu.cn (L. Chen).

http://dx.doi.org/10.1016/j.neucom.2015.02.091
0925-2312/

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 detail.

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, the 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

$$m_o\ddot{p} - m_o g = f_o$$
$$I_o\dot{\omega} + \omega \times I_o\omega = \tau_o \quad (1)$$

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, respectively, f_o and τ_o are the force and torque applied to the mass center of the object, respectively, 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]:

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

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/orientations 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

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

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:

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

where

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

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

Property 3 (Ge et al. [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:

$$\Gamma(t) = \int_0^\infty c(s)\,\mathrm{d}s \quad (6)$$

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

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

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 is 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