Proceedings of the 41st Chinese Control Conference
July 25-27, 2022, Hefei, China

Q-Learning Based Optimal Tracking Control of Free-Flying Space Manipulators with Unknown Dynamics

Hongxu Zhu[1], Shufan Wu[1], Qiang Shen[1], Ran Sun[1]

1. School of Aeronautics and Astronautics, Shanghai Jiaotong University, Shanghai 200240, P. R. China
E-mail: hcwsdxuxu@163.com; shufan.wu@sjtu.edu.cn; qiangshen@sjtu.edu.cn; sunr1990@163.com

Abstract: This paper investigates a Q-learning based optimal tracking problem of a free-flying space manipulator with unknown dynamics. First, via a series of equivalent transformations, we obtain a simplified model-free solution form of the original optimal tracking problem. Then we construct a Q-learning based policy iteration mechanism with an actor/critic neural network structure to solve the optimal control policy. By following a Lyapunov analysis we prove the asymptotical stability of the closed-loop system. Finally, a numerical simulation of a 2-DOF space manipulator verifies the effectiveness of the proposed method.

Key Words: Space manipulators, Linear quadratic tracking, Q-learning, Unknown dynamics

1 Introduction

The importance of robotics on-orbit servicing operations has been generally admitted with the development of space exploration [1]. A space manipulator system is used for various complex space missions, such as repairing, refueling, etc [2]. For such on-orbit servicing operations, one important step is to control the n-DOF robot manipulator to move toward the desired position such that the tool mounted at the end should touch the target and perform the follow-up tasks. To ensure the smooth implemention of the subsequent tasks, the end actuators of the robotic manipulator are required to track the reference command to a specified position with a specified ratation rate. That is to say, the space manipulator control problem is essentially a trajectory tracking problem for a continuous-time system.

Space manipulator systems have two cases: free-flying case and free-floating case. Compared to the latter, free-flying space manipulator systems mount the robotic arm on a thruster-equipped spacecraft platform so that they can perform larger tip displacements and keep the specific orientation. Due to its linearizability, investigations on tracking control of manipulators with ground fixed base [3–5] can be applied to free-flying manipulator systems in theory. However, traditional tracking control methods require exact information on system matrices, while those essential physical parameters may not be accurately measured. Thus, a model-free tracking control method is urgent needed. Moreover, the amount of fuel allocated to the space manipulator is very limited. How to optimize the energy consumption is also worth considering. Conventional optimal control policy design is often associated with the Hamilton-Jacobi-Bellman (HJB) equation. Unfortunately, to solve the HJB equation, accurate knowledge of the system dynamics is the necessary condition [6]. Even if one can estimate the dynamical information, the sequential step, solving the noncausal differential Riccati equations, is still hard to conduct online.

Recently, the applications of adaptive dynamic programming (ADP) have made significant progress in combining optimal control with adaptive control. In [7], a novel ADP based online optimal tracking control method without model dynamics is proposed aiming at continuous-time linear system. This provides an idea to solve the linear quadratic tracking problem without solving noncausal Riccati equations. In [8], a reinforcement learning control strategy is applied to the trajectory tracking on a flexible two-link manipulator system while suppressing vibration. In [9], single critic network instead of the aforementioned actor/critic network is used for the design of reinforcement learning-based control for robotic manipulators. It also concerns the input saturation problem. However, in those studies, stability analysis and policy convergence analysis are separated. In other word, no stability analysis about the closed-loop system with the reinforcement learning scheme has been conducted. This pheonomenon is quite common in reinforcement learning control, probably because the stability analysis for the entire closed-loop system is not tractable. In [10], an integral reinforcement learning (IRL) based model-free linear optimal control method is proposed. In particular, that work provides an idea to analyze the closed-loop asymptotical stability by regarding the actor/critic learning system as a multiscale system. This work has been further employed to event-triggered system [11] and motion planning [12–14], showing unlimited potential. However, that work mainly deals with linear regulation problems. How to extend the theory to optimal tracking problems and nonlinear systems requires further efforts. Motivated by those works, we proposed a model-free reinforcement learning based optimal tracking control method for the free-flying manipulator system. In this work, we will explore the stability analysis mechanism in the field of reinforcement learning based tracking control.

The main contributions of this paper have the following two aspects. First, a Q-learning based linear quadratic tracking controller is proposed for free-flying space manipulators which have unknown dynamics. The proposed method transforms the optimal tracking problem of the original system into the optimal control problem of the augmented system. Thus one can obtain the optimal control by solving one differential Riccati equation, which is conducted by IRL based policy iteration. Compared with [7, 10, 15], this method can cope with linear time-varing systems. Second, rigorous stability analysis is provided to guarantee that the state converges to the reference command, as well as the control policy converges to the optimal solution.

This work is supported by National Natural Science Foundation (NNSF) of China U20B2054.

The rest of this paper is organized as follows. Section 2 introduces the dynamic model and the optimal solution of linear quadratic tracking problems. Section 3 describes the theoretic formation of the proposed Q-learning based controller. Simulation example verifies the methodology in Section 4 and a brief conclusion is given in Section 5.

2 Problem Formulation and Preliminaries

Dynamical equations of a n-DOF free-flying rigid robotic manipulator can be described generally as follows

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

where $q := [q_1, q_2, \cdots, q_n]$ represents the joint positions of the manipulator, $u = [u_1, u_2, \cdots, u_n]^T$ is the execution torques on each joints of the manipulator, $M(q) \in \mathbb{R}^{n\times n}$ is a symmetric positive inertia matrix, $C(q,\dot{q})$ is the Coriolis and centripetal torque, and $u \in \mathbb{R}^n$ is the control input.

The system dynamics can be linearized when the space manipulator in free-flying case, which is

$$\dot{x} = Ax + Bu, \tag{2}$$

where $x = [q, \dot{q}]^T$, and

$$A = \begin{bmatrix} 0 & 1 \\ 0 & -M^{-1}(q)C(q,\dot{q}) \end{bmatrix}, B = \begin{bmatrix} 0 \\ M^{-1}(q) \end{bmatrix}.$$

Our task is to drive the joint positions to track the reference command, which is defined as $x_r = [q_d, \dot{q}_d]$, where $q_d = [q_{d1}, q_{d2}, \cdots, q_{dn}]$ and $\dot{q}_d = [\dot{q}_{d1}, \dot{q}_{d2}, \cdots, \dot{q}_{dn}]$. As this work focus on linear tracking problems, the reference command is assumed to be written in a linear model

$$\dot{x}_r = Fx_r, \tag{3}$$

where F is Hurwitz.

In order to quantitatively describe the tracking performance, we define the tracking error as $e = [e_1, e_2] = [q - q_d, \dot{q} - \dot{q}_d]$. To find the appropriate control input such that the closed-loop system is asymptotically stable and the tracking error is ultimately uniformly bounded, the performance index function reaches is defined as

$$J(t) = \frac{1}{2}\int_t^\infty \left[e^T M e + u^T R u\right]\, dt, \tag{4}$$

where $M \in \mathbb{R}^{2n\times 2n} \succeq 0$ and $R \in \mathbb{R}^{n\times n} \succ 0$ are user-defined symmetric matrices. Additionally, we assume that (A, B) is controllable and $(A, \sqrt{M})$ is observable.

In next part of this section, we will review the optimal solution for continuous-time linear quadratic tracking problem. Define the Hamiltonian function as follows

$$H_1 = \frac{1}{2}\left[(x - x_r)^T M(x - x_r) + u^T R u\right] + \lambda^T(Ax + Bu). \tag{5}$$

Let $\partial H_1/\partial u = 0$ and $\partial^2 H_1/\partial u^2 > 0$, the unconstrained optimal control can be written as follows

$$u = -R^{-1}B^T\lambda, \tag{6}$$

where λ satisfy the following costate equation

$$\dot{\lambda} = -\frac{\partial H_1}{\partial x} = -Mx + Mx_r - A^T\lambda. \tag{7}$$

Supposing that $\lambda = Px - g$, where $P \in \mathbb{R}^{2n\times 2n}$ and $g \in \mathbb{R}^{2n}$, we can obtain

$$\begin{aligned}\dot{\lambda} &= \dot{P}x + P\dot{x} - \dot{g} \\ &= \left[\dot{P} + PA - PBR^{-1}B^TP\right]x + PBR^{-1}B^Tg - \dot{g}.\end{aligned} \tag{8}$$

Substituting (8) into (7), we can get

$$\dot{P} + PA + A^TP - PBR^{-1}B^TP + M = 0, \tag{9}$$

$$\dot{g} + A^Tg - PBR^{-1}B^Tg + Mx_r = 0. \tag{10}$$

Now we concentrate on the optimal performance index on $[t, \infty]$, which is assumed to consist of quadratic term, primary term and constant term

$$J = \frac{1}{2}x^TPx - g^Tx + w. \tag{11}$$

To determine the constant term w, we consider the HJB equation, which is

$$\begin{aligned}-\frac{\partial J}{\partial t} =& \frac{\partial J}{\partial x^T}(Ax + Bu) + \frac{1}{2}e^TMe + \frac{1}{2}u^TRu \\ =& -x^T\dot{P}x + \dot{g}^Tx + \frac{1}{2}x_r^TMx_r - \frac{1}{2}g^TBR^{-1}B^Tg.\end{aligned} \tag{12}$$

Thus we derive the auxiliary function w of (11) satisfying

$$-\dot{w} = \frac{1}{2}x_r^TMx_r - \frac{1}{2}g^TBR^{-1}B^Tg. \tag{13}$$

Remark 1. The optimal tracking control (6) can be devided into two parts. The fisrt part is $u_1 = -R^{-1}B^TPx$, which can be regarded as the feedback gain, and the second part is $u_2 = R^{-1}B^Tg$, which can be regarded as the feed-forward gain. The optimal gain is the sum of those two gains, while in optimal regulation problems, the feed-forward gain does not exist. Different structures of the solution distinguish optimal tracking problems from optimal regulation problems.

Obviously, to obtain the optimal tracking control input, we need to solve both differential Riccati equation (9) and (10). The computation process itself is not tractable and con be only solved offline. Moreover, to solve (9) and (10), accurate information of system dynamics (i.e., knowledge of matrices $M(q)$ and $C(q,\dot{q})$) is required. Hence, directly acquire the optimal control policy is impractical.

3 Q-Learning for Optimal Tracking Control

In this section, a Q-learning based policy iteration method is proposed to approach the optimal linear tracking control without solving differential Riccati equations. In this reinforcement learning-based control structure, exact knowledge of system matrices is not required. Besides, we prove that the closed-loop system is asymptotically stable and tracking error is ultimately uniformly bounded.

3.1 Simplified Control Design for Linear Quadratic Tracking

Considering that the optimal tracking control gain consists of feedback and feed-foward gains, and the latter is related to the reference command, we define an augment state $X(t) = [x^T(t), x_r^T(t)]^T$. Then we can derive an augmented system including actual motion and reference command as

$$\dot{X} = \alpha X + \beta u, \tag{14}$$

where

$$\alpha = \begin{bmatrix} A & 0 \\ 0 & F \end{bmatrix}, \beta = \begin{bmatrix} B \\ 0 \end{bmatrix}.$$

For the augmented system (14), the performance index function (4) can be rewritten as

$$J(t) = \frac{1}{2}\int_t^\infty \left[X^T M_X X + u^T R u\right], \tag{15}$$

where

$$M_X = \begin{bmatrix} M & -M \\ -M & M \end{bmatrix}.$$

Let $S_{11} = P$, $S_{12}x_r = -g$, $S_{21} = S_{12}^T$ and $(1/2)x_r^T S_{22} x_r = w$, then the optimal performance index (11) can be rewritten as

$$\begin{aligned} J =& \frac{1}{2}x^T S_{11} x + x^T S_{12} x_r + \frac{1}{2}x_r^T S_{22} x_r \\ =& \frac{1}{2}X^T S X, \end{aligned} \tag{16}$$

where

$$S = \begin{bmatrix} S_{11} & S_{12} \\ S_{21} & S_{22} \end{bmatrix}.$$

Formula (16) seems like the structure of the optimal performance index of linear quadratic regulation problems. Then there is a question whether it is possible to construct a simplified solution structure of the optimal tracking control? The following Theorem gives the answer.

Theorem 1. For the linear system (2), the reference command is given by (3). Then the optimal tracking control policy is

$$u = -R^{-1}\beta^T S X, \tag{17}$$

where S satisfies the following Riccati equation

$$\dot{S} + \alpha^T S + S\alpha + M_X - S\beta R^{-1}\beta^T S = 0. \tag{18}$$

Proof. Expanding (17), we have

$$\begin{aligned} u =& -R^{-1}\begin{bmatrix} B^T & 0 \end{bmatrix}\begin{bmatrix} S_{11} & S_{12} \\ S_{21} & S_{22} \end{bmatrix}\begin{bmatrix} x \\ x_r \end{bmatrix} \\ =& -R^{-1}B^T P x + R^{-1}B^T g. \end{aligned}$$

From Section 2 we know (17) is the optimal tracking control policy. Then we try to show the rationality of (18). The upper left-hand side of (18) is

$$\dot{S}_{11} + A^T S_{11} + S_{11}A + M - S_{11}BR^{-1}B^T S_{11} = 0, \tag{19}$$

which is equivalent to (9). The upper right-hand side of (18) is

$$\dot{S}_{12} + A^T S_{12} + S_{12}F - M - S_{11}BR^{-1}B^T S_{12} = 0. \tag{20}$$

Multiplying x_r to the right-hand side of (20), we can obtain

$$\begin{aligned} &-\dot{S}_{12}x_r - A^T S_{12}x_r - S_{12}F x_r + M x_r \\ &+ S_{11}BR^{-1}B^T S_{12}x_r = 0, \end{aligned} \tag{21}$$

which is equivalent to (10). The lower right-hand side of (18) is

$$\dot{S}_{22} + F^T S_{22} + S_{22}F + M - S_{21}BR^{-1}B^T S_{12} = 0. \tag{22}$$

Multiplying $-x_r^T$ to the both left-hand and right-hand side of (22), we can obtain

$$\begin{aligned} &x_r^T \dot{S}_{22} x_r + x_r^T F^T S_{22} x_r + x_r^T S_{22} F x_r + x_r^T M x_r \\ &- x_r^T S_{21} B R^{-1} B^T S_{12} x_r = 0, \end{aligned} \tag{23}$$

which is equivalent to (13). This completes the proof.

3.2 IRL Based Policy Iteration

In reinforcement learning framework, one need to construct approximators to estimate the value function or performance index function and the optimal control policy. For Q-learning, the value function is an action-dependent value function, whose inputs include both current state and control policy. The action-dependent value function, also called Q-function, is user-defined. In deterministic cases, it follows

$$Q_t^*(x_t, u_t) = r_t + J_{t+\Delta t}^*(x_{t+\Delta t}, u_{t+\Delta t}), \tag{24}$$

where the subscript t represents the current moment, Δt represents a small time interval, the superscript $*$ represent the minimum value and r_t is the instant reward.

Although the Q-function is user-defined, its optimal value equals to the optimal performance index function. Thus, combining (16) with (19), (21) and (23), the Q-function is

$$\begin{aligned} Q =& \frac{1}{2}x^T S_{11} x + x^T S_{12} x_r + \frac{1}{2}x_r^T S_{22} x_r \\ &+ \frac{1}{2}x^T\left(\dot{S}_{11} + A^T S_{11} + S_{11}A + M \right. \\ &\left. - S_{11}BR^{-1}B^T S_{11}\right)x + x^T\left(\dot{S}_{12}x_r \right. \\ &A^T S_{12}x_r S_{12}F x_r - M x_r \\ &\left. - S_{11}BR^{-1}B^T S_{12}x_r\right) + \frac{1}{2}\left(x_r^T \dot{S}_{22} x_r \right. \\ &+ x_r^T F^T S_{22} x_r + x_r^T S_{22} F x_r + x_r^T M x_r \\ &\left. - x_r^T S_{21} B R^{-1} B^T S_{12} x_r\right) \\ =& \frac{1}{2}U^T \begin{bmatrix} Q_{xx} & Q_{xr} & Q_{xu} \\ Q_{rx} & Q_{rr} & Q_{ru} \\ Q_{ux} & Q_{ur} & Q_{uu} \end{bmatrix} U \\ :=& \frac{1}{2}U^T \overline{Q} U, \end{aligned} \tag{25}$$

where $U = [x^T\ x_r^T\ u^T]^T$, $Q_{xx} = S_{11} + \dot{S}_1 1 + A^T S_{11} + S_{11}A + M$, $Q_{xr} = Q_{rx}^T S_{12} + \dot{S}_{12} + A^T S_{12} + S_{12}F - M - S_{11}BR^{-1}B^T S_{11}$, $Q_{rr} = S_{22} + \dot{S}_{22} + F^T S_{22} + S_{22}F + M - S_{21}BR^{-1}B^T S_{12}$, $Q_{xu} = Q_{ux}^T = S_{11}B$, $Q_{ru} = Q_{ur}^T = S_{21}B$, $Q_{uu} = R$.

Obviously, we can reconstruct the structure of optimal tracking control in a model-free version by using the component of $\overline{Q}$, which is

$$u^* = -Q_{uu}^{-1}[Q_{ux}\ Q_{ur}][x^T\ x_r^T]^T. \tag{26}$$

Since the minimum of Q-function is equal to the optimal performance index function $J^*(t)$, which has following property according to (15),

$$J_{t+\Delta t} = J_t - \frac{1}{2}\int_t^{t+\Delta t}\left(X^T M_X X + u^{*T} R u^*\right) d\tau, \tag{27}$$

we can derive the following recursive fomula

$$Q_t^* = Q_{t+\Delta t}^* - \frac{1}{2}\int_t^{t+\Delta t} (X^T M_X X + u^{*T} R u^*)\, d\tau. \quad (28)$$

This is the IRL Bellman equation for continuous-time Q-learning. Policy iteration contains two steps, Q-funtion update and policy improvement. By solving (28), one can update the Q-function. Then we can update the policy by solving the following equation

$$\hat{u}_t = \arg\min_{\hat{u}} Q_t. \quad (29)$$

3.3 Learning-Based Control with Actor/Critic NN Structure

In this work, we employ an actor-critic NN structure to execute the policy iteration of Q-learning. The critic neural network is used to estimate the Q-function, which is

$$Q = W_c^T (U \otimes U), \quad (30)$$

where $W_c \in \mathbb{R}^{\frac{1}{2}3n\times(3n+1)}$ is the ideal weights of the critic neural network, $\otimes$ is a Kronecker-like operator which collects the terms and sorts in a unique order.

It is noted that (30) is strictly true as it is a variation on (25). The optimal tracking control is proposed as

$$\begin{aligned} u^* =& -\operatorname{mat}_n[W_c(\frac{1}{2}(7n^2+3n+1) : \frac{1}{2}(9n^2+3n))]^{-1} \\ & \cdot \operatorname{mat}_{2n\times n}[W_c(\frac{1}{2}(5n^2+n+1) : \frac{1}{2}(7n^2+3n))]^T \\ & \cdot [x^T\ x_r^T]^T \\ :=& -R^{-1}\overline{W}_c^T X, \end{aligned} \quad (31)$$

where $\operatorname{mat}_{a\times b}(\cdot)$ represents a user-defined matrization which transforms vector $A \in \mathbb{R}^{ab}$ into a matrix of dimension $a \times b$. Particularly, if $a = b$, it can be simplified as $\operatorname{mat}_a(\cdot)$.

However, ideal weights W_c are not obtainable without exact information of system dynamics. Thus the key to estimate the Q-function is approaching the ideal weights W_c by employing the self-learning technique to get estimated Q-function $\hat{Q}$ and actual weights of critic neural network $\hat{W}_c$. Then, we have

$$\hat{Q} = \hat{W}_c^T (U \otimes U). \quad (32)$$

Similarly, we employ an actor neural network to approximate the optimal tracking policy, which is

$$\hat{u} = \hat{W}_a^T X, \quad (33)$$

where $\hat{W}_a$ is the estimator of the ideal weights W_a given by

$$W_a = -R^{-1}\overline{W}_c^T$$

.

To describe the gap between the ideal weights and estimated weights, we define the following indicators

$$E_c = \frac{1}{2} e_c^T e_c, \quad (34)$$

and

$$E_a = e_a^T e_a, \quad (35)$$

where e_c and e_a are corresponding to critic NN and actor NN, which satisfy

$$\begin{aligned} e_c :=& \hat{Q}_{t+\Delta t} - \hat{Q}_t + \frac{1}{2}\int_t^{t+\Delta t} (X^T M_X X + u^T R u)\, d\tau \\ =& \hat{W}_c^T (U_{t+\Delta t} \otimes U_{t+\Delta t} - U_t \otimes U_t) \\ & + \frac{1}{2}\int_t^{t+\Delta t} (X^T M_X X + u^T R u)\, d\tau, \end{aligned} \quad (36)$$

and

$$e_a := \hat{W}_a^T X + R^{-1}\hat{\overline{W}}_c^T X. \quad (37)$$

Those two indicators E_c and E_a are always non-negative and turned to zero if and only if the ideal weights equal their estimates. In practice, to ensure the policy iteration stop in finite steps, a small threshold is applied to be an index. One can stop policy iteration if E_c and E_a drive down to the given threshold. Referring to the gradient descent method, we design the following tuning law for critic and actor estimators

$$\dot{\hat{W}}_c = -\alpha_c \frac{\partial E_c}{\partial \hat{W}_c} = -\alpha_c \delta e_c, \quad (38)$$

$$\dot{\hat{W}}_a = -\alpha_a X e_a^T, \quad (39)$$

where $\delta := U_{t+\Delta t} \otimes U_{t+\Delta t} - U_t \otimes U_t$, and $\alpha_c, \alpha_a \in \mathbb{R}$ are constant gains which influence the convergence rate.

Lemma 1. For any given control input $u(t)$, the estimated critic weights $\hat{W}_c$ converge to the ideal values W_c if the signal δ is persistently exciting (PE) over the interval $[t, t+T_{PE}]$ with $\int_t^{t+T_{PE}} \delta^T \delta \geq \sigma I$ with T_{PE} the excitation period and $\sigma > 0$.

Proof. Define the critic error $\tilde{W}_c := W_c - \hat{W}_c$. The critic error dynamics can be written as

$$\dot{\tilde{W}}_c = -\alpha_c \delta \delta^T \tilde{W}_c, \quad (40)$$

Considering the following Lyapunov function

$$\mathcal{L} = \frac{1}{2\alpha_c}\tilde{W}_c^T \tilde{W}_c, \quad (41)$$

we have

$$\dot{\mathcal{L}} = -\tilde{W}_c \delta \delta^T \tilde{W}_c. \quad (42)$$

According to the PE condition, $\dot{\mathcal{L}} \leq 0$ and the equality holds only if $\tilde{W}_c = 0$. Thus the origin is a stable equilibrium point for the critic error dynamical system (40). That is to say, estimated critic weights $\hat{W}_c$ are asymptotically convergent to the ideal weights W_c. This completes the proof.

3.4 Stability Analysis

Theorem 2. Consider a linearizable free-flying manipulator system given by (2) with the reference command given by (3). The approximator of Q-function and the optimal tracking policy are given by (32) and (33), respectively. The critic tuning law is given by (38) and the actor tuning law is given by (3.3). Then for all initial condition, the origin is a stable equilibrium point for the closed-loop system with the state $Z := [(x - x_r)^T\ \tilde{W}_c^T\ \tilde{W}_a^T]^T$ where $\tilde{W}_a = W_a - \hat{W}_a$, provided the following inequality holds,

$$1 < \alpha_a < \frac{\underline{\lambda}(R)}{\zeta}\left[\underline{\lambda}(M + \overline{W}_c R^{-1}\overline{W}_c^T) - \overline{\lambda}(\overline{W}_c \overline{W}_c^T)\right], \quad (43)$$

where ζ is a constant of unity order.

Proof. Consider the following Lyapunov function

$$L = \frac{1}{2}X^T S X + \frac{1}{2}\tilde{W}_c^T \tilde{W}_c + \frac{1}{2}\text{tr}\left[\tilde{W}_a^T \tilde{W}_a\right], \tag{44}$$

where tr$(\cdot)$ is the trace of a matrix. Differiating (44) leads to

$$\begin{aligned}\dot{L} =& X^T S(\alpha X + \beta\hat{u}) + \frac{1}{2}X^T \dot{S} X \\ &- \alpha_c \tilde{W}_c \delta\delta^T \tilde{W}_c + \text{tr}\left\{\tilde{W}_a^T \tilde{W}_a\right\}.\end{aligned} \tag{45}$$

Let $L_1 = X^T S(\alpha X + \beta\hat{u}) + \frac{1}{2}X^T \dot{S} X$, $L_2 = -\alpha_c \tilde{W}_c \delta\delta^T \tilde{W}_c$, $L_3 = \text{tr}\left(\tilde{W}_a^T \tilde{W}_a\right)$. By introducing (17) and (18), we can derive

$$\begin{aligned}L_1 =& X^T S\left(\alpha X + \beta u^* - \beta\tilde{W}_a^T X\right) + \frac{1}{2}X^T \dot{S} X \\ \leq& -\frac{1}{2}\left[\underline{\lambda}(M_X + \overline{W}_c R^{-1}\overline{W}_c^T) - \overline{\lambda}(\overline{W}_c \overline{W}_c^T)\right]\|X\|^2 \\ &+ \frac{1}{2}\|\tilde{W}_a^T X\|^2.\end{aligned} \tag{46}$$

The last inequality holds according to Young's inequality.

The dynamics of the actor weights can be written as

$$\dot{\hat{W}}_a = -\alpha_a X e_a^T = -\alpha_a X X^T \hat{W}_a - \alpha_a X X^T \hat{\overline{W}}_c R^{-1}. \tag{47}$$

Thus we derive the dynamics of estimation errors of $\hat{W}_a$ as

$$\dot{\tilde{W}}_a = -\alpha_a X X^T \tilde{W}_a - \alpha_a X X^T \tilde{\overline{W}}_c R^{-1}. \tag{48}$$

Since $\alpha_c \gg \alpha_a$, the convergence rate of $\hat{W}_c$ is far faster than $\hat{W}_a$. Thus

$$\begin{aligned}L_3 =& -\alpha_a \text{tr}\left[\tilde{W}_a X X^T \tilde{W}_a + \tilde{W}_a^T X X^T \tilde{\overline{W}}_c R^{-1}\right] \\ \leq& -\frac{\alpha_a}{2}\|\tilde{W}_a^T X\|^2 + \alpha_a \zeta\overline{\lambda}(R^{-1})\|X\|^2,\end{aligned} \tag{49}$$

where ζ is a constant of unity order. Then we can finally get

$$\begin{aligned}\dot{L} \leq& -\frac{1}{2}\left[\underline{\lambda}(M_X + \overline{W}_c R^{-1}\overline{W}_c^T) - \overline{\lambda}(\overline{W}_c \overline{W}_c^T)\right. \\ &\left.-\frac{\alpha_a}{2}\zeta\overline{\lambda}(R^{-1})\right]\|X\|^2 + \frac{1}{2}\|\tilde{W}_a^T X\|^2 \\ &- \frac{\alpha_a}{2}\|\tilde{W}_a^T X\|^2 - \alpha_c \tilde{W}_c^T \delta\delta^T \tilde{W}_c.\end{aligned} \tag{50}$$

The time derivative $\dot{L}$ is negative if the condition (43) holds. Therefore, the closed-loop system with the state $Z := [x^T\ x_r^T\ \tilde{W}_c^T\ \tilde{W}_a^T]^T$ is asymptotically stable.

4 Simulation

To verify the effectiveness of the proposed Q-learning based model-free tracking control method, we conduct a simulation for a 2-DOF space manipulator with the parameters shown in Table 1.

The dynamics of the free-flying spaces manipular is given by (1), the system matrices is given as follows

$$M(q) = \begin{bmatrix} M_{11} & M_{12} \\ M_{21} & M_{22} \end{bmatrix},\ C(q,\dot{q}) = \begin{bmatrix} C_{11} & C_{12} \\ C_{21} & C_{22} \end{bmatrix},$$

Table 1: Parameters of the Manipulator

Parameter	Description	Value
$m_1(kg)$	mass of link 1	1.0
$m_2(kg)$	mass of link 2	1.2
$m_e(kg)$	mass of tool	1.0
$I_1(kg \cdot m^2)$	inertia of link 1	0.12
$I_2(kg \cdot m^2)$	inertia of link 2	0.14
$I_e(kg \cdot m^2)$	inertia of tool	0.10
$l_1(m)$	length of link 1	1.0
$l_2(m)$	length of link 2	1.2
$l_e(m)$	length of tool	0.5
$l_{c1}(m)$	mass center of link 1	0.5
$l_{c2}(m)$	mass center of link 2	0.6
$l_{ce}(m)$	mass center of tool	0.3

where $M_{11} = 2a_1\cos(q_2) + 2a_2\sin(q_2) + a_3$, $M_{12} = M_{21} = a_1\cos(q_2) + a_2\sin(q_2) + a_4$, $M_{22} = a_4$, $C_{11} = -(a_1/sinq_2 - a_2\cos q_2)\dot{q}_2$, $C_{12} = -(a_1\sin(q_2) - a_2\cos(q_2))((\dot{q})_1 + \dot{q}_2)$, $C_{21} = (a_1\sin(q_2) - a_2\cos(q_2))\dot{q}_1$, $C_{22} = 0$, $a_1 = (m_e l_{ce}\cos(\delta_e) + m_e l_2 + m_2 l_{c2})l_1$, $a_2 = -m_e l_1 l_{ce}\sin(\delta_e)$, $a_3 = I_1 + I_2 + I_e + m_c l_{c1}^2 + m_2(l_1^2 + l_{c2}^2) + m_e(l_1^2 + l_2^2 + 2l_2 l_{ce}\cos(\delta_e) + l_{ce}^2)$, $a_4 = I_2 + I_e + m_2 l_{c2}^2 + m_e(l_2^2 + 2l_2 l_{ce}\cos(\delta_e) + l_{ce}^2)$.

The main task of this simulation is drive the joint position of the manipulator to the desired position $q_r = [0\ 0]^T$ and rate $\dot{q}_d = [0\ 0]^T$ from a reasonable initial state, which is set as $q = [30\deg\ -30\deg]^T$ and $\dot{q} = [10\deg/\text{s}\ 0\deg/\text{s}]$. The angle between the tool and link 2 denoted as δ_e is 30deg. The convergence rate relevant parameters are defined as $\alpha_c = 50$ and $\alpha_a = 5$. The step size in this work is set as $\Delta t = 0.01$s.

In order to demonstrate the superiority of the proposed algorithm, we adopt a constant control gain to regulate the space manipulator, which is

$$K = \begin{bmatrix} 17.1042 & 3.6634 & 9.9975 & 0.2248 \\ 3.5218 & 12.0511 & -0.2248 & 9.9975 \end{bmatrix}.$$

The control gain is obtained by solving the LQR problem of an approximate linear parameter invariant (LPI) dynamic model of the manipulator.

The trajectories of the joint positions and rates conducted by the constant gain are shown in Fig.1, while the trajectories conducted by proposed Q-learning-based control algorithm are presented in Fig.2. Although the actual states of two groups are convergent to the reference command, the control performance is different. Obviously, comparing with the constant gain, the proposed Q-learning based control scheme can change the gain to adaptively match the current dynamic system such that the trajectories have shorter convergence time and fewer oscillations. Fig.3 and Fig.4 show the evolutions of the execution torques of the constant gain and proposed algorithm, respectively. We can conclude that the proposed control algorithm requires less energy.

5 Conclusion

In this paper, a model-free Q-learning based linear quadratic tracking control method is developed and applied to free-flying space manipulators. By transforming the original optimal tracking problem into the optimal stabilizing control problem of the augmented system, which simplifies the control structure. Then we design a rational action-

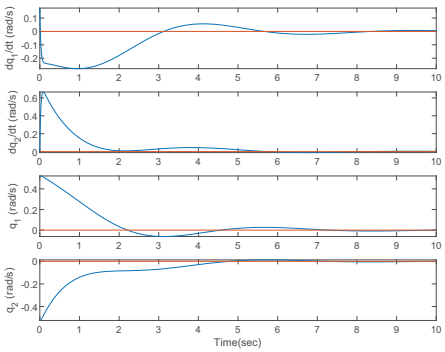

Fig. 1: Positions and rates of the manipulator joints by constant control gain

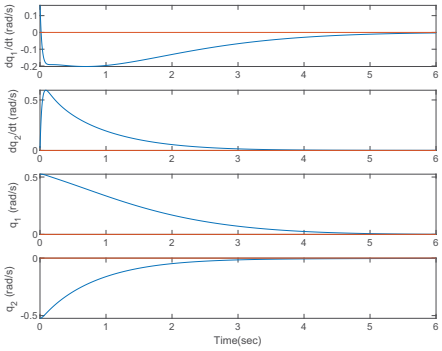

Fig. 2: Positions and rates of the manipulator joints by proposed Q-learning based control algorithm

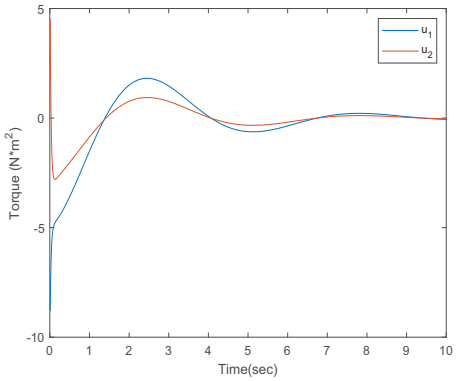

Fig. 3: Execution torques by constant control gain

dependent function and derive the IRL Bellman equation. Based on that we construct a policy iteration approach to generate the optimal control policy and an actor/critic neural network structure is employed to the actual execution. To explain the rationality of such design we prove the asymptotical stability of the closed-loop system. Finally, the numerical example verify the validity of the proposed Q-learning scheme. Due to space limitation, we do not conduct dynamic tracking simulations and consider the output tracking problems, which will be investigated in the future work.

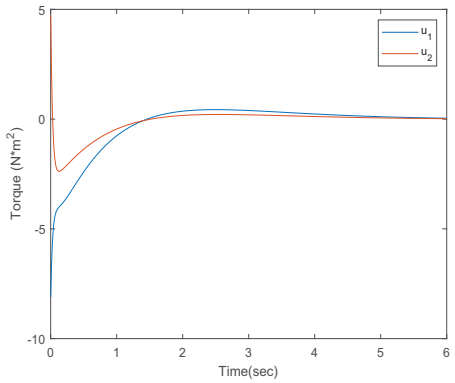

Fig. 4: Execution torques by proposed Q-learning based control algorithm

References

[1] A.Flores-Abad, O. Ma, K. Pham, and S.Ulrich, A review of space robotics technologies for on-orbit servicing, *Progress in Aerospace Sciences*, 68: 1–26, 2014.

[2] NASA. On-orbit satellite servicing study project report. Technical Report, NASA, 2010.

[3] A. Liu, H. Zhao, T. Song, Z. Liu, H. Wang, and D. Sun, Adaptive control of manipulator based on neural network, *Neural Computing and Applications*, 33(9):4077-4085, 2021.

[4] P. Gierlak, and M. Szuster, Adaptive position/force control for robot manipulator in contact with a flexible environment, *Robotics and Autonomous Systems*, 95:80-101, 2017.

[5] M. Spong, On the robust-control of robot manipulators, *IEEE Transactions on Automatic Control*, 37(11):1782-1786, 1992.

[6] F. Lewis, D. Vrabie, and V. Syrmos, *Optimal control*. New York: Wiley, 2012.

[7] C. Qin, H. Zhang, and Y. Luo, Online optimal tracking control of continuous-time linear systems with unknown dynamics by using adaptive dynamic programming, *Internation Journal of Control*, 87(5):1000-1009, 2014.

[8] W. He, H. Gao, C. Zhou, C. Yang, Z. Li, Reinforcement learning control of a flexible two-link manipulator: an experimental investigation, *IEEE Trans. Systems, Man, and Cybernetics: Systems*, 51(12):7326-7336, 2021.

[9] G. Cheng, and L. Dong, Optimal Control for Robotic Manipulators With Input Saturation Using Single Critic Network, *2019 Chinese Automation Congress*, 2019: 2344-2349.

[10] K. Vamvoudakis, Q-learning for continuous-time linear systems: A model-free infinite horizon optimal control approach, *Systems & Conttol Letters*, 100:14-20, 2017.

[11] K. Vamvoudakis, and H. Ferraz, Model-free event-triggered control algorithm for continuous-time linear systems with optimal performance, *Automatica*, 87:412-420, 2018.

[12] G. Kontoudis, and K. Vamvoudakis, Kinodynamic motion planning with continuous-time Q-Learning: an online, model-free, and safe navigation framework, *IEEE Trans. Neural Networks and Learning Systems*, 30(12):3803-3817, 2019.

[13] Y. Yang, K. Vamvoudakis, H. Modares, and W. He, Safe intermittent reinforcement learning for nonlinear systems, *2019 IEEE Conference on Dicision and Control*, 2019: 690-697.

[14] C. Sun, and K. Vamvoudakis, Continuous-time safe learning with temporal logic constraints in adversarial environments, *2020 American Control Conference*, 2020: 4786-4791.

[15] W. He, H. Huang, and S. Ge, Adaptive neural network control of a robotic manipulator with Time-varying output constraints, *IEEE Trans. Cybernetics*, 47(10):3136-3147, 2017.