Actor–critic learning based PID control for robotic manipulators

Hamed Rahimi Nohooji[a], Abolfazl Zaraki[b], Holger Voos[a,c]

[a]*Interdisciplinary Centre for Security, Reliability and Trust (SnT), Automation Robotics Research Group, University of Luxembourg*
[b]*Robotics Research Group, University of Hertfordshire, UK*
[c]*Faculty of Science, Technology and Medicine (FSTM), Dept. of Engineering, University of Luxembourg*

Abstract

In this paper, a reinforcement learning structure is proposed to auto-tune PID gains by solving an optimal tracking control problem for robot manipulators. Taking advantage of the actor-critic framework implemented by neural networks, optimal tracking performance is achieved while unknown system dynamics are estimated. The critic network is used to learn the optimal cost-to-go function while the actor-network converges it and learns the optimal PID gains. Furthermore, Lyapunov's direct method is utilized to prove the stability of the closed-loop system. By that means, an analytical procedure is delivered for a stable robot manipulator system to systematically adjust PID gains without the ad-hoc and painstaking process. The resultant actor-critic PID-like control exhibits stable adaptive and learning capabilities, while delivered with a simple structure and inexpensive online computational demands. Numerical simulation is performed to illustrate the effectiveness and advantages of the proposed actor-critic neural network PID control.

Keywords: Reinforcement learning, Actor-critic, PID control, Neural network, Robot manipulators

1. Introduction

Energy conservation and environmental protection are becoming increasingly important due to the progress of society and the limitation of energy resources. To deliver energy-efficient systems, the focus of control algorithms

is shifting to low-energy control developments. Optimal control that aims to fulfill the control task by consuming the least control source has gained increasing importance in modern technologies [27]. This method is emerging as one of the fundamental tools in recent dynamical systems studies, driven by practical needs coupled with the ability to overcome theoretical challenges [34, 11, 16, 42, 21, 40].

Real-world engineering systems have continuous nonlinear features. The solution of nonlinear optimal control depends on the solution of the underlying Hamilton–Jacobi-Bellman (HJB) equation [50]. But, the HJB equation is a first-order nonlinear partial differential equation that is very difficult to solve [59]. Heretofore, finding an exact analytic solution for such intractable nonlinear equations has remained an unsolved problem. One practical solution is linearizing the system. Riccati equation is a typical method to cope with the optimal control of a linear system [6]. This method can provide an exact solution using analytical approaches. In our previous work, we studied an extension of the Riccati equation, named inverse differential Riccati equation, to deliver a closed-loop optimal control for a fixed-end-point linear system over a specific time interval [42]. Moreover, several researchers employed numerical approaches to solve the nonlinear optimal control [8, 43, 22].

Instead of delivering a direct approach to solve the HJB equation, several algorithms were developed to approximate the solution of nonlinear optimal controls. Policy iteration [15] involves a computational intelligence technique that can evaluate, and then improve the cost of a control policy until converging to the optimal controller. Generalized policy iteration [54] is a family of optimal learning techniques which has policy iteration at one extreme. Dynamic programming [5] is another method proposed to optimize nonlinear control. But, this method works in a backward-in-time manner and thus suffers from the problem of "dimensional curse", and computational untenability. Adaptive dynamic programming [33] is a forward-in-time online learning algorithm based on value iteration. This method is based on an actor-critic framework in which the critic network is used for value function approximation while an actor-network is employed for approximation of the control policy [26, 31, 28].

Reinforcement Learning (RL) [49] can be formulated based on value-based methods, policy gradient methods, or actor-critic methods, for nonlinear control problems [17, 61, 46, 57]. In actor-critic RL, typically the critic network evaluates the performance of the control policy, and the actor-network generates and improves the controlling action sequence in the system [12, 38, 62].

Most available actor-critic RL optimal controls for nonlinear systems require the exact model acknowledged [50, 9]. However, driving the exact model for real-world nonlinear physical systems is often impossible, leading such controls to be inefficient in practical engineering implementation. Accordingly, recent studies are tending toward developing RL optimal controls for systems with unknown nonlinear dynamics. Nonetheless, to compensate for the dynamics uncertainty, many of those developments require the utilization of an adaptive approximation-based identifier [58] or an observer [60], which increases the computation complexity. Most importantly, almost all of those algorithms to ensure delivering stable performance, required complicated control with different gains to be tuned [7, 39]. This may result in complex yet unreliable control which is extremely dependent on extra steps for gain tuning/estimating.

Proportional–integral–derivative (PID) controls have been known as one of the most practical tools in engineering applications. It offers a simple yet efficient solution to many real-world control problems due to its simplicity and intuitiveness in both structure and concept [13, 48]. Control gain determination is the key to PID design. For classical PID controllers, the gains typically remain constant during the execution. It may degrade the overall performance of the closed-loop system. Thus, to function satisfactorily, PID parameters have to be properly designed and tuned [35]. Several techniques like fuzzy logic [51, 52, 14], neural network [1], genetic algorithm, [67] or particle swarm optimization [19, 45] were developed to present tuning of PID gains. However, integrating such approaches to control design, increases the complexity of the overall controller. Also, such controllers suffer from stability guarantees, while stability has always been a great concern with engineering systems since uncertain dynamics or disturbances are prone to drive the system unstable.

Motivated by the above discussion, we propose a simple actor-critic RL optimal tracking control for nonlinear robot manipulators. The core of the proposed control is based on a PID structure with the minimum required gains to be determined. Actor-critic RL with neural networks is used through the direct Lyapunov analysis to auto-tuning the PID gains. Bearing the PID structure, the proposed control is simple in structure, inexpensive in computation, and easy to implement. On the other hand, by taking advantage of RL framework learning properties, the optimal performance of the system is addressed. Accordingly, superior tracking control with improved performance is achieved.

Compared with the existing literature, the main advantages of this study are summarized as follows.

1. This work combines Lyapunov-based PID control with actor-critic RL to achieve a simple, stable optimal tracking control solution for nonlinear robot manipulators. Accordingly, stable optimal performance is achieved using an efficient while simple framework. Thus, this study delivers energy-efficient and practical control frameworks thanks to its intuitiveness in concept and simplicity in design.
2. Different from most PID controls suffering an ad-hoc and painstaking process to determine PID gains, the proposed method delivers a systematic way to obtain such gains with the minimum required parameters to determine. Direct Lyapunov analysis combined with the learning-based scheme and neural networks is utilized to automatically and continuously update the gains. This feature allows the dynamics nonlinearities and uncertainties to be addressed. As a result, a structurally simple self-tuning approach is achieved, which ensures system stability and guarantees prescribed performance specifications.
3. In contrast to many publishing RL-based optimal control methods like [66], and [2], our presented actor-critic frameworks do not require model knowledge. Also, it removes the need for employing an identifier [63] or observer [30, 41] to compensate for the unknown or uncertain system dynamics. On top of this, inspired by the behavior of the generalized PID error signal, we introduced a novel cost function that can properly improve the tracking performance. All make the proposed approach practical for real-world nonlinear engineering systems.

The rest of the paper is structured as follows. Section 2 delivers preliminaries and formulates the control problem. Section 3 provides the actor-critic reinforcement PID control design and analyzes the stability of the system. Simulations are illustrated to show the effectiveness of the proposed control framework in Section 4. Section 5 discussed the advantages of the method and the future direction of the paper. Finally, the conclusion has summarized the paper in Section 6.

Notations. Throughout this paper, we use $\mathbb{R}$ to denote the sets of real numbers. $(\tilde{\bullet}) = (\hat{\bullet}) - (\bullet^*)$, with $(\bullet^*)$, and $(\hat{\bullet})$, are indicated the optimal, and the estimated values of $(\bullet)$, respectively. Vertical bars $\|\bullet\|$ stand with the Frobenius norm for matrices or the Euclidean norm for vectors, and, $\lambda_{\max}(\bullet)$

and $\lambda_{\min}(\bullet)$ represent the largest and the smallest eigenvalues of a square matrix $(\bullet)$, respectively.

Note that throughout this article, for simplifying notation, the arguments in variables or functions are dropped, whenever possible, if no confusion is likely to occur.

2. Problem Formulation and Preliminaries

2.1. Problem Formulation

Consider the dynamic model of a robot manipulator as

$$M\ddot{q} + C\dot{q} + G = \tau, \tag{1}$$

where $M(q) \in \mathbb{R}^{n\times n}$ denotes the mass matrix and $q(t) \in \mathbb{R}^n$ is the generalized joint coordinate vector with n number of joints, $C(q,\dot{q}) \in \mathbb{R}^n$ represents the centrifugal and Coriolis forces vector, $G(q) \in \mathbb{R}^n$ is the vector of gravitational forces/torques, and $\tau(t) \in \mathbb{R}^n$ is the vector of generalized torques acting at the joints.

Property 1 [24], [47]. The mass matrix M is symmetric and positive definite. In addition, the matrix $\dot{M} - 2C$ is skew-symmetric, i.e., $\nu^T\left(2C - \dot{M}\right)\nu = 0$, for all $\nu \in \mathbb{R}^n$.

Property 2 [47], [25]. The mass matrix M is positive definite, and its matrix norm is bounded by $\psi_m > 0$, and $\psi_M > 0$ such that $\psi_m \leq \|M\| \leq \psi_M$. Furthermore, for some unknown positive constants χ and γ, $\|C\| \leq \chi\|\dot{q}\|$, $\|G\| \leq \gamma$.

Note that the subsequent development is based on the assumption that joint coordinate vector $q(t)$, and its first-time derivative $\dot{q}(t)$ are measurable, and dynamics gains $M(q)$, $C(q,\dot{q})$, and $G(q)$ are unknown.

The objective of the paper is to design reinforcement learning-based PID control law for the robot system (1) such that the system uncertainties are accommodated adaptively, the robot joint position signal $q(t)$ moves along a given desired trajectory $q_d(t)$, as closely as possible, and all the internal signals are guaranteed to be bounded while minimizing a desired cost-to-go

function. To this end, the assumption below is imposed.

Assumption 1. The desired trajectory $q_d(t) \in \mathbb{R}^n$, and its first and second-time derivatives, i.e., $\dot{q}_d(t) \in \mathbb{R}^n$, and $\ddot{q}_d(t) \in \mathbb{R}^n$, respectively, are continuous, bounded and accessible in real-time to the controller.

2.2. Preliminary

A neural network approximation is used to approximate the system uncertainties and the cost function, and to determine PID gains through a direct Lyapunov method. Based on its powerful learning abilities, we employ Radial Basis Function (RBF) neural network [32, 65] to approximate continuous function $f(Z) : \mathbb{R}^m \to \mathbb{R}$, as $f(Z) = \omega^T h(Z)$, where $Z \in \Omega_z \subset \mathbb{R}^m$ is the neural network input vector with m being the neural network input dimension, $\omega \in \mathbb{R}^r$ is the neural network weight vector with r is the node number, $h(Z) = [h_1(Z), h_2(Z), \ldots, h_r(Z)]^T$ is a vector of basis function vector with $h_i(Z)$ being the Gaussian functions for $i = 1, ..., r$, and expressed as $h_i(Z) = \exp\left(-(Z-\alpha_i)^T(Z-\alpha_i)\big/\beta^2\right)$, with $\alpha_i = [\alpha_{i1}, \alpha_{i2}, \ldots, \alpha_{im}]^T$ being the center of the i^{th} input element of the neural network, and β being the width of the Gaussian functions. In [44], it has been shown that by choosing sufficient number of nodes, the RBF neural network can approximate any continuous function $f(Z)$ over the compact set $\Omega_z \subset \mathbb{R}^m$, as $f(Z) = \omega^{*^T} h(Z) + \varepsilon(Z), \ \forall Z \in \Omega_z \subset \mathbb{R}^m$, where ω^* is the ideal constant weight vector, and $\varepsilon(Z)$ is the unknown approximation error which is upper bounded in the sense that $\|\varepsilon(Z)\| \leq \varepsilon_M$, $\forall Z, \in \Omega_z \subset \mathbb{R}^m$ with $\varepsilon_M \in \mathbb{R}^+$ being an unknown constant [10].

Lemma 1. *[23] For the Gaussian RBF neural network $f(Z) = \omega^T h(Z)$, there exists a constant $C_h > 0$ such that*

$$\|h(Z)\| \leq C_h, \tag{2}$$

where C_h is taken as $\sum_{k=0}^{\infty} 3m(k+2)^{m-1} \exp\left(-2\nu^2 k^2/\beta^2\right)$, and ν is being $\nu := \frac{1}{2}\min_{i\neq j}\|\alpha_i - \alpha_j\|$.

Remark 1. *It has been shown in [56] that since the infinite series*

$$\left\{3m(k+2)^{m-1} \exp\left(-2\nu^2 k^2/\beta^2\right)\right\}, (k = 0, 1, \ldots, +\infty)$$

is convergent by the Ratio Test Theorem [3], the upper bound C_h in (2) is a limited value. Also, it is clear that C_h is independent of the neural network input variables, Z, and the dimension of neural weights, r.

Remark 2. *A number of parameters including lower or upper bounds are defined by Property 2, Lemma 1, Remark 1, and in defining the neural networks. These bounds will be used to formulate the control frameworks and to analyze the system's stability. However, although these parameters exist, they will not involve in designing the control. Accordingly, actual estimation of them will not be required in setting up and implementing the control scheme.*

Lemma 2. *[36] [29] Consider a positive function given by*

$$V(t) = \sum_{i=1}^{n} \frac{1}{2} e_i(t)^T Q(t) e_i(t)$$

with $Q(t) = Q^T(t) > 0$ is a dimensionally compatible matrix, and initial bounded condition $V(0)$. If the following inequality holds:

$$\dot{V}(t) \leq -\iota_1 V(t) + \iota_2, \tag{3}$$

where ι_1, and ι_2 are positive constants, then, the error signal e_i in the closed-loop system remain in the compact set $\Omega_r := \{\Upsilon | \, \|\Upsilon\| \leq \aleph_r\}$, and they will finally converge to the convergence compact set $\Omega_c := \{\Upsilon | \, \|\Upsilon\| \leq \aleph_c\}$, where $\aleph_r = \sqrt{2\left(V(0) + \iota_2/\iota_1\right)}$, and $\aleph_c = \sqrt{2\iota_2/\iota_1}$.

Lemma 3. *[18, 37] Consider*

$$\Xi(t) = 2\upsilon e(t) + \upsilon^2 \int_0^t e(t)\, d\rho + de(t)/dt,$$

with $\upsilon > 0$, and $e(t) \in \mathbb{R}^n$. Then, the boundedness of $\Xi(t)$ guaranties the boundedness $e(t)$, $\int_0^t e(t)\, d\rho$, and $de(t)/dt$.

3. Control Design

This work introduces an actor-critic reinforcement learning framework into a PID control design of robot manipulators. The critic neural network approximates the cost function, and the actor neural network tunes the PID gains to generate the control signal. The schematics of the proposed actor-critic learning-based PID control system are shown in Fig.1.

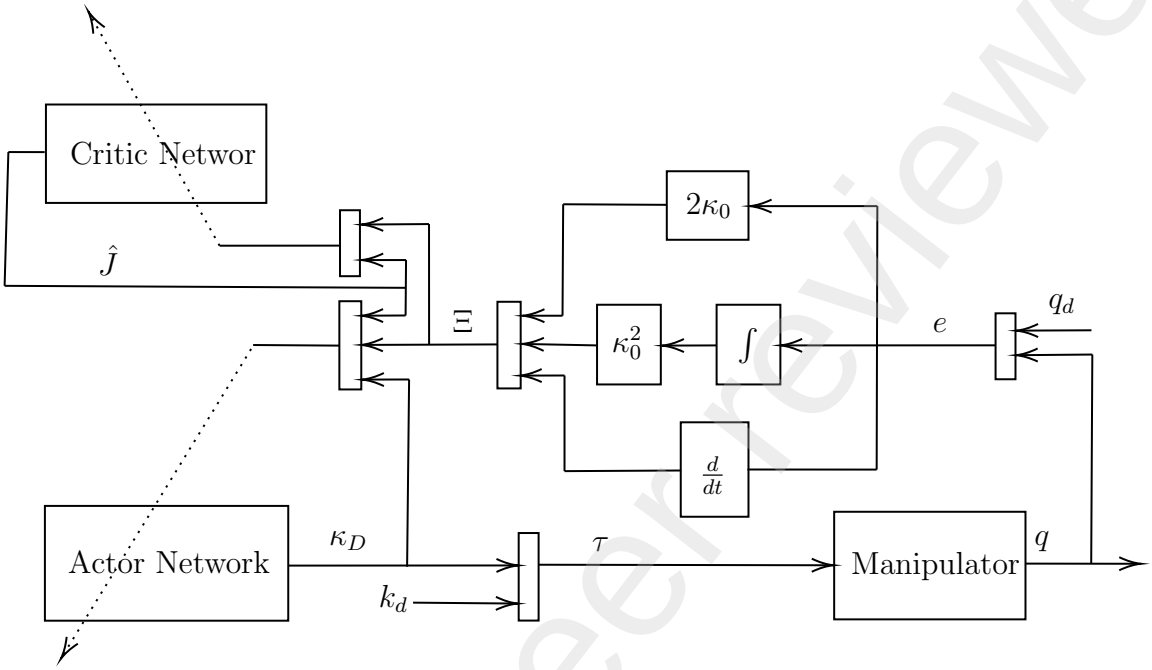


Figure 1: The architecture of the proposed actor-critic learning based PID control system

3.1. Reinforcement Learning

3.1.1. Critic Network

A long-term discounted cost is defined as,

$$J(t) = \int_t^{\infty} e^{-\frac{m-t}{\psi}} r(m)\, dm, \tag{4}$$

where ψ is a time constant for discounting the future cost.
Inspiring by the intuitiveness of PID control in the tracking dynamical systems, i.e., canceling the steady-state error, boosting the closed-loop system dynamics, and thus improving the tracking of low-frequency references [4], we introduce the instant cost function as

$$r(t) = \Xi(t)^T Q \Xi(t) + \tau(t)^T R \tau(t), \tag{5}$$

with $\Xi(t)$ is a PID-like generalized error signal which defines as $\Xi(t) = 2\upsilon e(t) + \upsilon^2 \int_0^t e(t)\, d\rho + \upsilon de(t)/dt$, with $e(t) = q_d(t) - q(t)$, υ is a positive constant, and Q and R are positive *semi-definite* matrices that could not be zero at the same time. Note that the cost function (5) is different

from that in the traditional LQR problem, in which *first,* $\Xi(t)$ is not only the error or its first-time derivation but also includes the integration of the error in the execution time which can inclusively improve the tracking performance; and *second,* R is a positive semi-definite matrix. Thus, in the case that strictly precise tracking is required, one can set $R = 0$ to more emphasis on tracking performance.

Let $J = W_c^{*T} h_c(Z_c) + \varepsilon_c$, and $\hat{J} = \hat{W}_c^T h_c(Z_c)$ with $Z_c = \Xi$, and define the prediction error as

$$\begin{aligned} \delta(t) &= r(t) - \frac{1}{\psi}\hat{J}(t) + \dot{\hat{J}}(t) \\ &= r(t) + \hat{W}_c{}^T \left(\dot{h}_c(Z_c) - \frac{1}{\psi} h_c(Z_c) \right) = r(t) + \hat{W}_c{}^T \Lambda, \end{aligned} \tag{6}$$

where $\Lambda = \dot{h}_c - h_c/\psi$. Note that (6) is similar to the Hamiltonian defined in [53], and [55]. Consider $E_c = 1/2\delta^T\delta$, then ideal approximation is achieved if error function E_c is minimized. Utilizing the gradient descent method, the updating law for the critic network is designed as

$$\dot{\hat{W}}_c = -\sigma \frac{\partial E_c}{\partial \hat{W}_c} = -\sigma \left(r + \hat{W}_c^T \Lambda \right) \Lambda, \tag{7}$$

where $\sigma > 0$ is the learning rate for the critic network. Ultimately, the so-called $\sigma -$ modification term is added to the critic network updating law (7), to ensure the boundedness of $\| \hat{W}_c \|$, and to improve the robustness of the closed-loop system [38, 68, 20]. Then, (7) is formed as

$$\dot{\hat{W}}_c = -\sigma \left(r + \hat{W}_c^T \Lambda \right) \Lambda - \sigma \eta_c \hat{W}_c, \tag{8}$$

where η_c is a positive constant.

3.1.2. Actor Network

Consider a positive definite Lyapunov function candidate as

$$V_r = \frac{1}{2} \Xi^T M \Xi. \tag{9}$$

Note that, according to Lemma 3, the boundedness of the generalized error $\Xi(t)$, ensures the boundedness of error signal $e(t)$. Thus, the original tracking control task can boil down to stabilizing $\Xi(t)$. Accordingly, by choosing the

Lyapunov function (9), and developing a systematic strategy to set the control input $\tau(\cdot)$, such that Ξ is bounded, we can achieve the control objective on bounding the error signal.

Differentiation V_r with respect to time gives

$$\dot{V}_r = \Xi^T M \dot{\Xi} + \frac{1}{2}\Xi^T \dot{M}\Xi. \tag{10}$$

Considering the definition of the generalized error Ξ, and using the robot dynamics (1), we obtain $M\dot{\Xi} = C\dot{q} + G - \tau + M\digamma(\cdot)$, where $\digamma(\cdot) = \ddot{q}_d + 2\kappa_0\dot{e} + \kappa_0^2 e$, with $e(t) = q_d(t) - q(t)$ is a computable term. Since all variables are accessible, then, considering Properties 1 and 2 and utilizing Young's inequality [64], we have

$$\begin{aligned}
\Xi^T C\dot{q} &\leq \alpha\|\Xi\|^2\chi^2\|\dot{q}\|^4 + 1/4\alpha,\\
\Xi^T G &\leq \alpha\|\Xi\|^2\gamma^2 + 1/4\alpha,\\
\Xi^T M\digamma &\leq \alpha\|\Xi\|^2\psi_M^2\|\digamma\|^2 + 1/4\alpha,\\
\frac{1}{2}\Xi^T\dot{M}\Xi = \Xi^T C\Xi &\leq \alpha\|\Xi\|^2\chi^2\|\dot{q}\|^2\|\Xi\|^2 + 1/4\alpha,
\end{aligned}$$

where $\alpha > 0$ is a design parameter. Accordingly, considering (10) and utilizing the above inequalities, we can obtain

$$\begin{aligned}
\dot{V}_r &\leq \alpha\|\Xi\|^2\left(\chi^2\|\dot{q}\|^4 + \gamma^2 + \mu^2\|\digamma\|^2 + \chi^2\|\dot{q}\|^2\|\Xi\|^2\right) - \Xi^T\tau + \frac{1}{4\alpha}\\
&\leq -\alpha\|\Xi\|^2\Gamma - \Xi^T\tau + \frac{1}{\alpha},
\end{aligned} \tag{11}$$

with $\Gamma = -\left(\chi^2\|\dot{q}\|^4 + \gamma^2 + \mu^2\|\digamma\|^2 + \chi^2\|\dot{q}\|^2\|\Xi\|^2\right)$.

We propose PID-like control input as

$$\tau = \left(k_p + \kappa_P(\cdot)\right)e(t) + \left(k_i + \kappa_I(\cdot)\right)\int_0^t e(t)d\rho + \left(k_d + \kappa_D(\cdot)\right)\frac{de(t)}{dt}, \tag{12}$$

with k_p, k_i, and k_d are positive constant. As it is clear in (12), compared with the traditional PID control that involves only constant gains, our proposed PID-like control includes time-varying gains $\kappa_P(\cdot)$, $\kappa_I(\cdot)$, and $\kappa_D(\cdot)$. On that basis, we have six degrees of freedom to select parameters independently. To

reduce the complexity of the gain tuning parameters, we employed coefficient $\upsilon > 0$ to link the gain's parameters and rewrite them in terms of derivative gains as: $k_d = k_p/2\upsilon = k_i/\upsilon^2$; and similarly, $\kappa_D(\cdot) = \kappa_P(\cdot)/2\upsilon = \kappa_I(\cdot)/\upsilon^2$. Accordingly, by designing υ so that $e^2 + 2\upsilon e + \upsilon^2$ is Hurwitz, we can reform (12) to only require two parameters to determine, i.e., k_d, and $\kappa_D(\cdot)$. Then, the PID control is expressed as

$$\tau = (k_d + \kappa_D(\cdot)) \left(2\upsilon e(t) + \upsilon^2 \int_0^t e(t)\, d\rho + \frac{de(t)}{dt} \right). \tag{13}$$

Accordingly, the complex task of determining PID gains is reduced to choosing two constants k_d and υ, plus determining $\kappa_D(\cdot)$, which will tune automatically using the actor-network updating law as will explain in the following.

In this work, we estimate the time-varying PID gain κ_D using neural networks as $\kappa_D = -\alpha \hat{W}_a^T h_a(Z_a)$, where α is a constant gain, and $Z_a = \left[q^T, \dot{q}^T, e^T, \dot{e}^T, \digamma^T, \Xi^T\right]$ is the input vector of actor-network. Let $\tilde{W}_a = \hat{W}_a - W_a^*$, and define the instant estimation error $\xi_a = \tilde{W}_a^T h_a$. The objective of the updating law of the actor-network is to minimize the estimation error ξ_a and the estimated cost function $\hat{J}$, and to improve the tracking performance. Accordingly, by defining the actor-network integrated error as $\varsigma_a = \xi_a + \Xi^T\Xi + k_a\hat{J}$, where $k_a > 0$ is a control gain, and minimizing the actor-network error function

$$E_a = \frac{1}{2}\varsigma_a^T \varsigma_a, \tag{14}$$

using the gradient descent method, the updating law of actor-network $\hat{W}_a$ can be obtained as

$$\begin{aligned} \dot{\hat{W}}_a = -\alpha \frac{\partial E_a}{\partial \hat{W}_a} &= -\alpha \frac{\partial E_a}{\partial \varsigma_a} \frac{\partial \varsigma_a}{\partial \xi_a} \frac{\partial \xi_a}{\partial \hat{W}_a} \\ &= -\alpha \left(\xi_a + \Xi^T\Xi + k_a\hat{J} \right) h_a. \end{aligned} \tag{15}$$

Moreover, since the actual value of ξ_a is unavailable, we use its estimation as $\hat{\xi}_a$ [38, 68], and further adding a $\sigma -$ modification term, e.g., $\alpha\eta_a\hat{W}_a$, where $\eta_a > 0$, and eventually modify the updating law (15) as

$$\dot{\hat{W}}_a = -\alpha \left(\hat{W}_a^T h_a + \Xi^T\Xi + k_a\hat{J} \right) h_a - \alpha\eta_a\hat{W}_a. \tag{16}$$

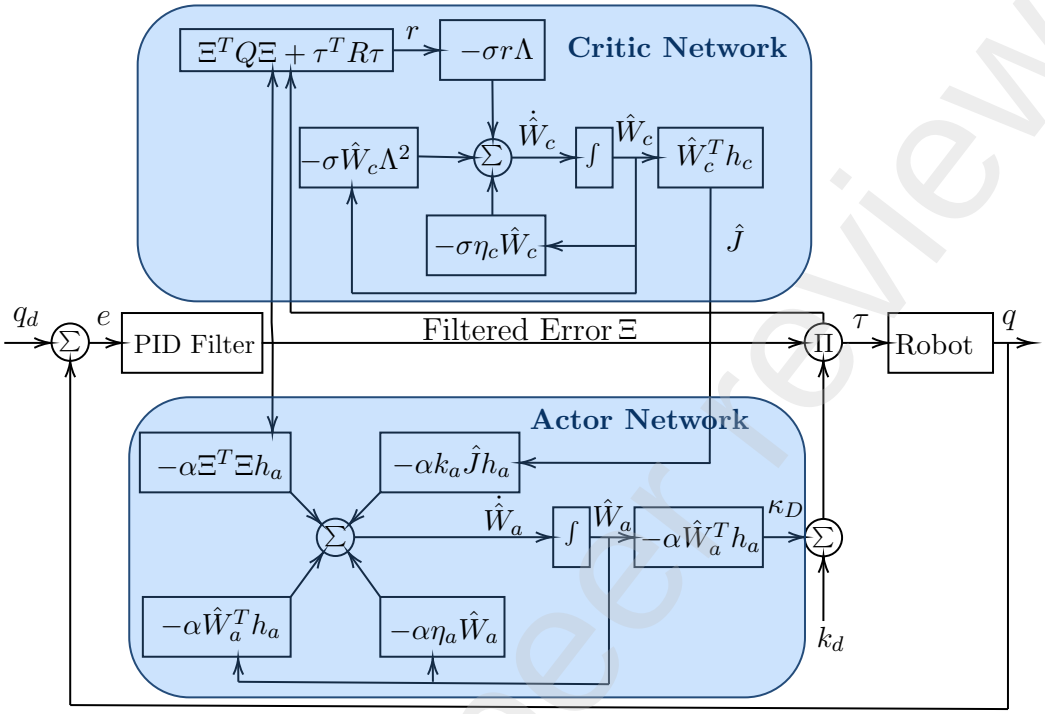


Figure 2: The overall proposed actor-critic PID control for robot manipulator system

The detail of the proposed actor–critic PID control structure is shown in Fig. 2.

3.2. Stability Analysis

Theorem 1. Consider the robot dynamics (1), with Properties 1 and 2. Suppose Assumption 1 holds. Let the PID control input be given by (13). Let the critic network and actor-network updating rules be given by (8), and (16), respectively; then all the closed-loop signals are bounded.

Proof. Consider the Lyapunov function as

$$V = V_r + V_c + V_a, \tag{17}$$

where V_r is defined at (9), $V_c = 1/2\sigma^{-1}\tilde{W}_c^T\tilde{W}_c$, and $V_a = 1/2\tilde{W}_a^T\tilde{W}_a$. We utilize RBF neural networks in order to compensate for the system uncertainties as $\Gamma = {W_a}^{*T} h_a + \varepsilon_a$. Then, considering the definition of the generalized error

$\Xi(t)$, and applying the control (13), (11) becomes

$$\begin{aligned}\dot{V}_r &\leq -\alpha\|\Xi\|^2\Gamma - \Xi^T(k_d+\kappa_D)\Xi + \frac{1}{\alpha},\\ &\leq -\Xi^T\left(k_d+\alpha\varepsilon_a\right)\Xi + \alpha\|\Xi\|^2\tilde{W}_a^T h_a + \frac{1}{\alpha}.\end{aligned} \tag{18}$$

First-time derivation of V_c considering (8) leads to

$$\dot{V}_c = \sigma^{-1}\tilde{W}_c^T\dot{W}_c = -\tilde{W}_c\left(r + \hat{W}_c^T\Lambda\right)\Lambda - \eta_c\tilde{W}_c^T\hat{W}_c. \tag{19}$$

Since $r\left(t\right) = J/\psi - \dot{J}$, then

$$r = -W_c^{*T}\Lambda + e_c,$$

where $e_c = -\dot{\varepsilon_c} + \varepsilon_c/\psi$. Accordingly (19) can be rewritten as

$$\dot{V}_c = -\tilde{W}_c^T\left(-W_c^* + \hat{W}_c^T\right)\Lambda^T\Lambda - \tilde{W}_c^T\Lambda e_c - \eta_c\tilde{W}_c^T\left(\tilde{W}_c + W_c^*\right). \tag{20}$$

Applying the Young's inequality one has

$$-W_c^*\tilde{W}_c^T \leq \frac{1}{2}W_c^{*2} + \frac{1}{2}\tilde{W}_c^T\tilde{W}_c,$$

$$-\tilde{W}_c^T\Lambda e_c \leq \tilde{W}_c^T\Lambda\Lambda^T\tilde{W}_c + \frac{1}{4}e_c^2,$$

then, (20) can be rewritten as

$$\dot{V}_c \leq -\frac{\eta_c}{2}\tilde{W}_c^T\tilde{W}_c + \frac{\eta_c}{2}\|W_c^*\|^2 + \frac{1}{4}e_{cMax}^2, \tag{21}$$

where e_{cMax} is the upper bound of e_c.
First-time derivation of V_a considering (16) can be obtained as

$$\begin{aligned}\dot{V}_a &= \tilde{W}_a^T\dot{\tilde{W}}_a\\ &= -\alpha\tilde{W}_a^T\left[\left(\hat{W}_a^T h_a + \Xi^T\Xi + k_a\hat{J}\right)h_a + \eta_a\hat{W}_a\right]\\ &= -\alpha\tilde{W}_a^T h_a\hat{W}_a^T h_a - \alpha\tilde{W}_a^T h_a\Xi^T\Xi - \alpha\tilde{W}_a^T h_a k_a\hat{J} - \alpha\eta_a\tilde{W}_a^T\hat{W}_a.\end{aligned} \tag{22}$$

Considering

$$\tilde{W_a}^T\hat{W}_a = \tilde{W_a}^T\left(W_a^* + \tilde{W}_a\right) = \tilde{W_a}^T W_a^* + \tilde{W_a}^T\tilde{W}_a,$$

and applying the Young's inequality one has

$$\tilde{W_a}^T W_a^* \leq \left(\tilde{W_a}^T \tilde{W_a} + {W_a}^{*T} W_a^*\right)/2,$$

leading to

$$-\alpha\eta_a \tilde{W_a}^T \hat{W}_a \leq -\alpha\eta_a \left(\tilde{W_a}^T \tilde{W_a} - {W_a}^{*T} W_a^*\right)/2.$$

Moreover, considering $\hat{J} = W_c^* h_c \tilde{W}_c h_c$, then,

$$\hat{J}^T \hat{J} \leq 2(W_c^* h_c)^T W_c^* h_c + 2(\tilde{W}_c h_c)^T \tilde{W}_c h_c.$$

Furthermore, the following inequalities can be obtained by applying Young's inequality,

$$-\tilde{W}_a^T h_a \hat{W}_a^T h_a \leq -\left(\tilde{W}_a^T h_a\right)^2 + \tilde{W}_a^T h_a W_a^{*T} h_a \leq -\frac{1}{2}\left(\tilde{W}_a^T h_a\right)^2 + \frac{1}{2}\left(W_a^{*T} h_a\right)^2.$$

$$-\tilde{W}_a^T h_a k_a \hat{J} \leq \frac{1}{2}\left(\tilde{W}_a^T h_a\right)^2 + \frac{1}{2}k_a^2 \hat{J}^2 \leq \frac{1}{2}\left(\tilde{W}_a^T h_a\right)^2 + k_a^2\left(\tilde{W}_c^T h_c\right)^2 + k_a^2\left(W_c^{*T} h_c\right)^2$$

Combining with the above inequality, (22) can be rewritten as

$$\begin{aligned}\dot{V}_a \leq & -\alpha\tilde{W}_a^T h_a \Xi^T \Xi - \frac{\alpha}{2}\eta_a \tilde{W}_a^T \tilde{W}_a + \frac{\alpha}{2}\|h_a\|^2\|W_a^*\|^2 + \frac{\alpha}{2}\eta_a\|W_a^*\|^2 \\ & + \alpha k_a^2\|W_c^*\|^2\|h_c\|^2 + \alpha k_a^2\|h_c\|^2 \tilde{W}_c^T \tilde{W}_c.\end{aligned} \tag{23}$$

Recalling (18), (21), and (23), one can obtain

$$\begin{aligned}\dot{V} &= \dot{V}_r + \dot{V}_c + \dot{V}_a \\ &\leq -\Xi^T\left(k_d + \alpha\varepsilon_a\right)\Xi + \alpha\|\Xi\|^2 \tilde{W}_a^T h_a + \frac{1}{\alpha} - \frac{\eta_c}{2}\tilde{W}_c^T \tilde{W}_c + \frac{\eta_c}{2}\|W_c^*\|^2 + \frac{1}{4}e_{cMax}^2 \\ &- \alpha\|\Xi\|^2 \tilde{W}_a^T h_a - \frac{\alpha}{2}\eta_a \tilde{W}_a^T \tilde{W}_a + \frac{\alpha}{2}\left(\eta_a + \|h_a\|^2\right)\|W_a^*\|^2 \\ &+ \alpha k_a^2\|W_c^*\|^2\|h_c\|^2 + \alpha k_a^2\|h_c\|^2 \tilde{W}_c^T \tilde{W}_c \\ &= -\Xi^T\left(k_d + \alpha\varepsilon_a\right)\Xi - \frac{1}{2}\left(\eta_c - 2\alpha k_a^2\|h_c\|^2\right)\tilde{W}_c^T \tilde{W}_c - \frac{\alpha}{2}\eta_a \tilde{W}_a^T \tilde{W}_a \\ &+ \left(\frac{\eta_c}{2} + \alpha k_a^2\|h_c\|^2\right)\|W_c^*\|^2 + \frac{\alpha}{2}\left(\eta_a + \|h_a\|^2\right)\|W_a^*\|^2 + \frac{1}{4}e_{cMax}^2 + \frac{1}{\alpha}.\end{aligned} \tag{24}$$

According to Lemma 1, and as detailed in Remark 1, the bounding of the basis functions $\|h_c\|$, and $\|h_a\|$ are expressed as $\rho_{mc} \leq \|h_c\| \leq \rho_{Mc}$, and $\rho_{ma} \leq \|h_a\| \leq \rho_{Ma}$, respectively. Then, according to the definition of $V = V_r + V_c + V_a$, with $V_r = 1/2\Xi^T M \Xi$, $V_c = 1/2\sigma^{-1}\tilde{W}_c^T\tilde{W}_c$, and $V_a = 1/2\tilde{W}_a^T\tilde{W}_a$, then (24) can be rewritten as

$$\dot{V} \leq -\iota_1 V + \iota_2, \tag{25}$$

where ι_1, ι_2 are defined as follow:

$$\iota_1 = \min\left\{2\frac{\lambda_{\min}\left(k_d - \alpha\varepsilon_a\right)}{\lambda_{Max}\left(M\right)}, \eta_c - 2\alpha k_a^2 {\rho_{mc}}^2, \alpha\eta_a\right\},$$

$$\iota_2 = \frac{(\eta_c + 2\alpha k_a^2\rho_{Mc}^2)\omega_c^2}{2} + \frac{\alpha(\eta_a + \rho_{Ma}^2)\omega_a^2}{2} + \frac{e_{cMax}^2}{4} + \frac{1}{\alpha},$$

with ω_c and ω_a denote the upper bounds of optimal weights $\|W_c^*\|$, and $\|W_a^*\|$, respectively. Correspondingly, design parameters k_a, k_d, α, and η_c need to properly select to ensure $\iota_1 > 0$. Then, according to Lemma 2, Ξ, W_c and W_a remain semi-globally uniformly ultimately bounded. Furthermore, using Lemma 3, $e\left(t\right)$, $\int_0^t e\left(t\right)d\rho$, and $de\left(t\right)/dt$, are remained bounded as $\Xi\left(t\right)$ is bounded. Then, according to Assumption 1, $q\left(t\right)$ and $\dot{q}\left(t\right)$ are bounded. Also, since $\tilde{W}_a = \hat{W}_a - W_a^*$, and $\tilde{W}_c = \hat{W}_c - W_c^*$, then $\hat{W}_a$, and $\hat{W}_c$ are bounded. Finally, the boundedness of the above-mentioned signals and considering Lemma 1 which indicates the boundedness of the basis function vectors h_a, and h_c, leads to the boundedness of control $\tau(t)$ in (13), and eventually concluding the boundedness of all closed-loop signals. This completes the proof.

Remark 3. *Note that the existence of ι_2 in (25) reveals that the system only achieves stability, but it could not achieve exponential stability. For completeness, multiply (25) by $e^{\iota_1 t}$, then $\frac{d}{dt}\left(e^{\iota_1 t}V\right) \leq \iota_2 e^{\iota_1 t}$, and further integrate it, then*

$$V \leq \left(V(0) - \frac{\iota_2}{\iota_1}\right)\iota_2 e^{-\iota_1 t} + \frac{\iota_2}{\iota_1} \leq V(0) + \frac{\iota_2}{\iota_1}.$$

This implies the boundedness of the Lyapunov function $V(t)$, and further according to Lemma 2 indicates error signals Ξ, W_c and ς_a remain in the compact set defined by

$$\Omega_r := \left\{\Psi \middle| \, \|\Psi\| \leq \sqrt{2\left(V(0) + \iota_2/\iota_1\right)}\right\}$$

and will eventually converge to the compact sets defined by

$$\Omega_c := \left\{ \Psi | \, \|\Psi\| \leq \sqrt{2\iota_2/\iota_1} \right\}.$$

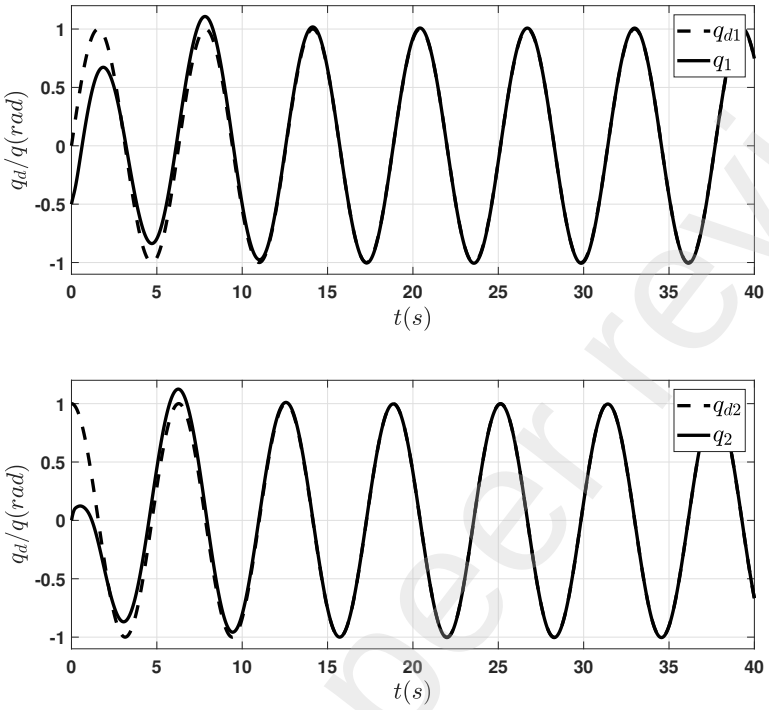
4. Illustrative example

In this section, numerical simulations are performed to verify the effectiveness of the proposed actor-critic learning-based PID control in Theorem 1. A simple two-link robot manipulator in the vertical plane is used for the simulation study. Physical robot parameters were chosen as follows: mass of the links $m_1 = m_2 = 5$kg, length of the links $l_1 = l_2 = 1$m. The desired trajectories are chosen as $q_d = [\sin(t); \cos(t)]$, and the initial condition of each joint is given by $q(0) = [-0.5; 0]$,$\dot{q}(0) = [1; 0]$. The cost function parameters are chosen as $Q = 100I$, and $R = 0.01I$, where I is the identity matrix. Control parameters chosen to be $\sigma = 50$, $\alpha = 50$, $k_d = 50$, $k_a = 0.1$, $\eta_c = 0.01$, $\eta_a = 0.01$, $\upsilon = 0.5$, and $\psi = 1000$.
To do the simulation study, the unknown dynamic model of the system is considered, and to approximate uncertainties, a radial basis function neural network with ten nodes on each hidden layer is chosen. For both the critic networks and actor networks centers α_i are evenly distributed in the span of input space $[-2.5, 2.5]$, and widths of $\beta = 1$. The starting points of neural networks weights were chosen as $\hat{W}_c(0) = \hat{W}_a(0) = 0$. Simulation results are shown in Figs. 3-7.

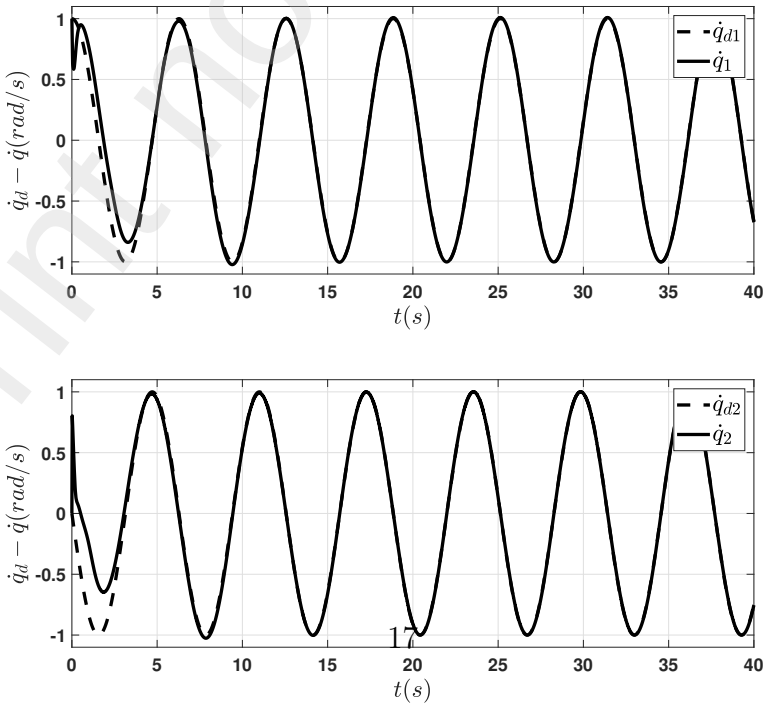
The tracking performances of links are shown in 3-5. Figures 3a, and 3b illustrate that the actual position and velocity signals can follow their desired trajectories. Figures 4a, and 4b demonstrate the position, and velocity errors, respectively. Figure 5 shows the generalized PID integrated error function. Figure 6 shows the boundedness of both critic and actor parameter vectors. Finally, input control is depicted in Fig. 7. From the above figures, it can be seen that *i*) the tracking errors converge to a small zero neighborhood, which implies that the robot's joint positions and velocities follow the desired signals; *ii*) input control and both critic and actor parameter vectors are bounded. Therefore, the proposed method can accomplish the control tasks.

5. Discussion

As technology advances, machine learning methods have been treated as an advantageous part of functional control systems. Optimal control, on the

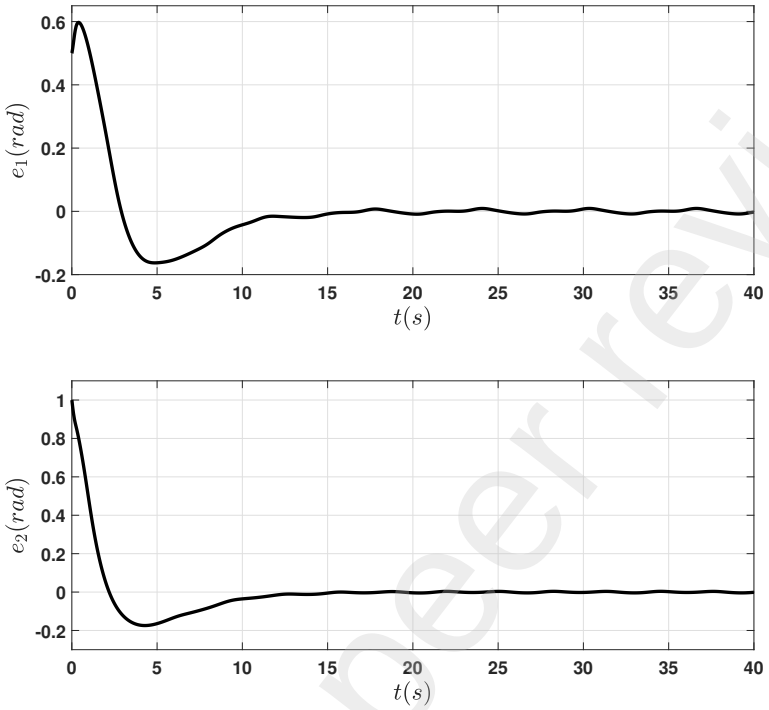


(a) Desired trajectories and actual trajectories of joint positions

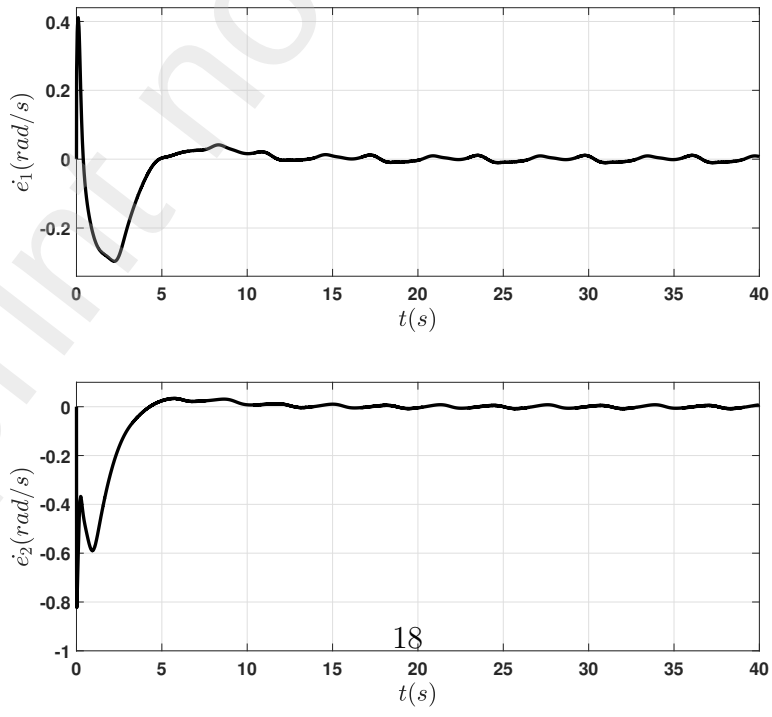


(b) Desired trajectories and actual trajectories of joint velocities

Figure 3: Position and velocity tracking



(a) Trajectories of position error



(b) Trajectories of velocity error

Figure 4: Tracking error

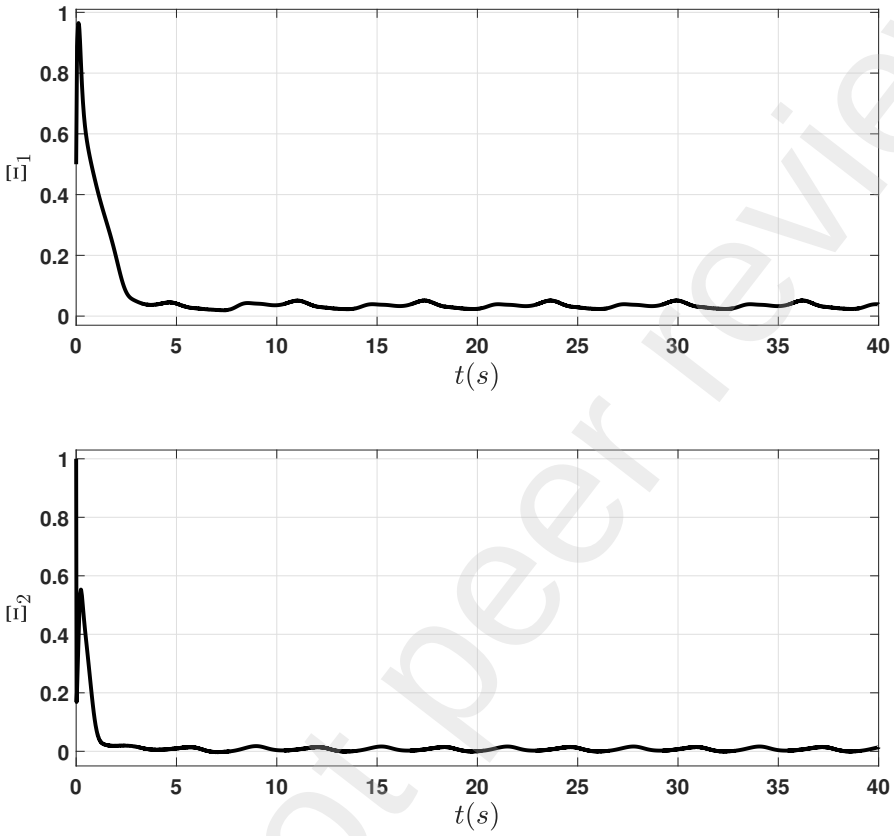
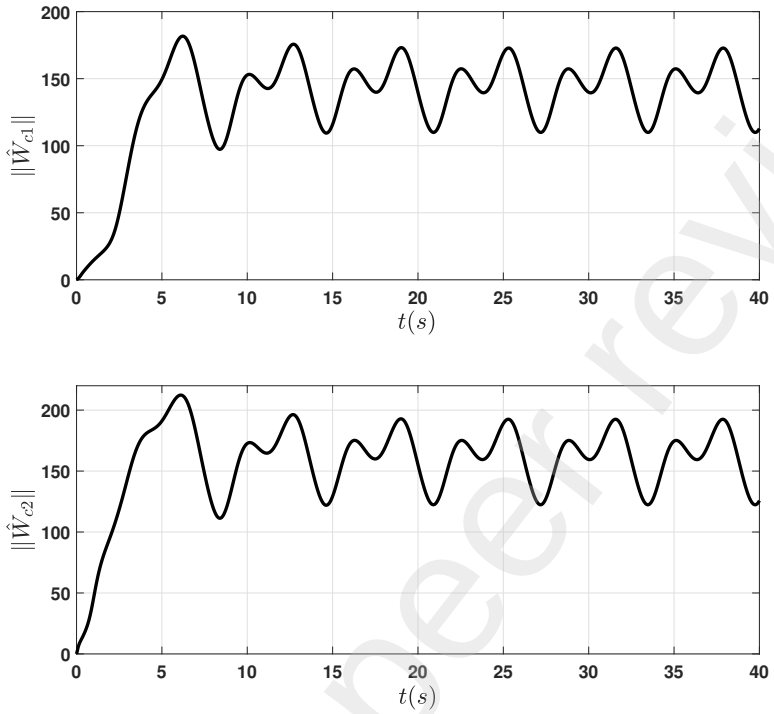
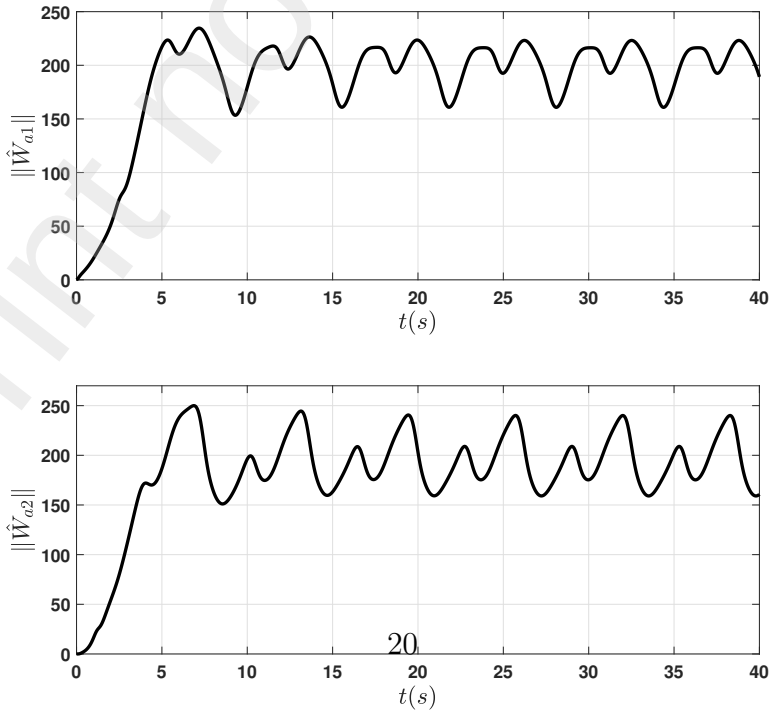


Figure 5: Trajectories of the PID integrated error function

other hand, gained significant importance in recent practical control systems, as it leads to controls with the least input source and thus meets the requirement of low-energy control design targets. Also, even though numerous advanced control frameworks have been developed for dynamical systems due to their simple structures and functional effectiveness, PID regulators remain the backbone of most practical control systems. The above discussion motivates our effort to establish a novel framework for constructing an optimal control structure and endow it with an actor-critic reinforcement learning mechanism capable of achieving the optimal solution. This development is delivered to a stable PID-like control for robot manipulators which is en-



(a) Norms of the critic network



(b) Norms of the actor network

Figure 6: Norms of the critic-actor network

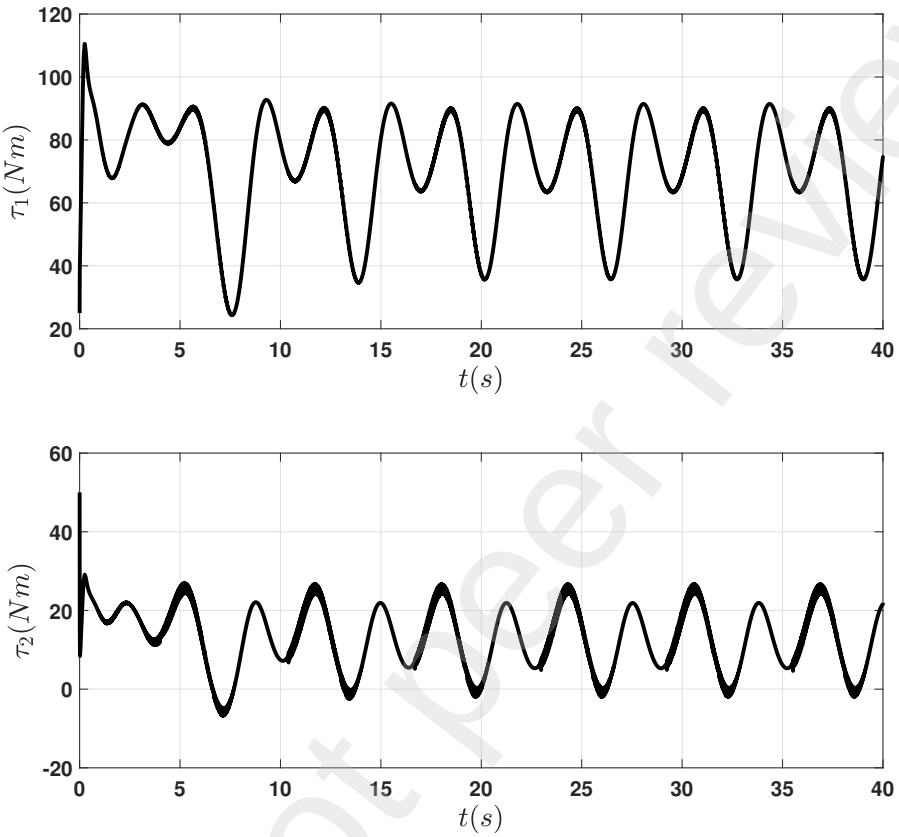


Figure 7: Trajectories of control input

riched by analytical development capable of systematically correcting PID gains.

Traditional PID controls are typically case-based. Also, they require hand-tuning mainly by trial and error practice. More importantly, their gains remain constant within the control process. In contrast, our proposed PID gains include time-varying parts which are tuned automatically during the entire system operation. Furthermore, different from most available PID controls, in which gains are selected/tuned independently, thanks to defining linking parameter υ, in our proposed PID-like control, the gains are determined correlatively with minimum requested parameters.

In contrast with several developed RL diagrams, which are suffering from the problem of learning behavior through trial-and-error interactions with a dynamic environment, advancing from online tuning weights algorithm, the presented actor-critic framework has distinguished advantages like shortening the learning time. Moreover, different from the most traditional RL control scheme which can not deliver system stability, utilizing the Lyapunov-based direct analysis, the analytical development is derived for the weight updating laws, and accordingly, the stability of the closed-loop system is guaranteed. The proposed learning-based tracking approach relaxed the requirement of explicit model knowledge, or linearly-in-parameter conditions in robot dynamics, making it effective in serving real-world research and engineering practice. Furthermore, as it is significantly simple and low-cost in implementation compared with available methods, it can realize to proper control framework for industrial robot manipulators.

Future work deals with other robotics scenarios like rehabilitative robotics and assistive robotics, where the interaction with the human should be considered. Furthermore, since the intelligent control of soft robotics has become a popular research topic, our next works will extend the work to soft actuators and robotics. Note that since the presented method utilized neural networks and reinforcement learning, it has good approximation properties to overcome major soft robotic challenges like the existence of infinite degrees of freedom.

6. Conclusion

In this paper, we proposed an actor-critic learning-based adaptive PID control method to address the optimal tracking problem of uncertain robot manipulators. We developed a new conceptual continuous-time performance index to evaluate tracking performance and control behavior, and employed a critic network to approximate it and deliver a reinforcement signal to the action network. The actor-network then developed a new low-complexity stable PID control with optimal self-tuning gain structure. Our numerical simulations demonstrate that the resultant control frameworks are functionally effective in controlling robotic manipulators.

Our study represents a significant contribution to the theoretical development of learning-based PID control for robotic manipulators, and paves the way toward the design of efficient and simple, yet energy-saving control principles for nonlinear dynamical systems. However, future work will involve experimental validations of our proposed control method in different robotic scenarios to further validate our findings.

References

[1] Akhyar, S., Omatu, S., 1993. Self-tuning pid control by neural-networks, in: Proceedings of 1993 International Conference on Neural Networks (IJCNN-93-Nagoya, Japan), IEEE. pp. 2749–2752.

[2] Al-Tamimi, A., Lewis, F.L., Abu-Khalaf, M., 2008. Discrete-time nonlinear hjb solution using approximate dynamic programming: Convergence proof. IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics) 38, 943–949.

[3] Apostol, T.M., Ablow, C., 1958. Mathematical analysis. Physics Today 11, 32.

[4] Åström, K.J., Murray, R.M., 2021. Feedback systems: an introduction for scientists and engineers. Princeton university press.

[5] Bellman, R., 1966. Dynamic programming. Science 153, 34–37.

[6] Bittanti, S., Laub, A.J., Willems, J.C., 2012. The Riccati Equation. Springer Science & Business Media.

[7] Chen, Q., Jin, Y., Song, Y., 2022. Fault-tolerant adaptive tracking control of euler-lagrange systems–an echo state network approach driven by reinforcement learning. Neurocomputing 484, 109–116.

[8] Diehl, M., Gerhard, J., Marquardt, W., Mönnigmann, M., 2008. Numerical solution approaches for robust nonlinear optimal control problems. Computers & Chemical Engineering 32, 1279–1292.

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

[10] Ge, S.S., Wang, C., 2002. Adaptive nn control of uncertain nonlinear pure-feedback systems. Automatica 38, 671–682.

[11] Geering, H.P., 2007. Optimal control with engineering applications. Springer.

[12] Grondman, I., Busoniu, L., Lopes, G.A., Babuska, R., 2012. A survey of actor-critic reinforcement learning: Standard and natural policy gradients. IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews) 42, 1291–1307.

[13] Hagglund, T., Astrom, K.J., 1995. Pid controllers: theory, design, and tuning. ISA-The Instrumentation, Systems, and Automation Society .

[14] Han, J., Shan, X., Liu, H., Xiao, J., Huang, T., 2023. Fuzzy gain scheduling pid control of a hybrid robot based on dynamic characteristics. Mechanism and Machine Theory 184, 105283.

[15] Howard, R.A., 1960. Dynamic programming and markov processes. .

[16] Hull, D.G., 2013. Optimal control theory for applications. Springer Science & Business Media.

[17] Kaelbling, L.P., Littman, M.L., Moore, A.W., 1996. Reinforcement learning: A survey. Journal of artificial intelligence research 4, 237–285.

[18] Khalil, H.K., 2000. Universal integral controllers for minimum-phase nonlinear systems. IEEE Transactions on automatic control 45, 490–494.

[19] Kim, T.H., Maruta, I., Sugie, T., 2008. Robust pid controller tuning based on the constrained particle swarm optimization. Automatica 44, 1104–1110.

[20] Kiumarsi, B., Lewis, F.L., 2014. Actor–critic-based optimal tracking for partially unknown nonlinear discrete-time systems. IEEE transactions on neural networks and learning systems 26, 140–151.

[21] Korayem, M., Haghpanahi, M., Rahimi, H., Nikoobin, A., 2009. Finite element method and optimal control theory for path planning of elastic manipulators, in: New Advances in Intelligent Decision Technologies. Springer, pp. 117–126.

[22] Korayem, M.H., Nohooji, H.R., Nikoobin, A., 2012. Mathematical modeling and trajectory planning of mobile manipulators with flexible links and joints. Applied Mathematical Modelling 36. doi:10.1016/j.apm.2011.10.002.

[23] Kurdila, A., Narcowich, F.J., Ward, J.D., 1995. Persistency of excitation in identification using radial basis function approximants. SIAM journal on control and optimization 33, 625–642.

[24] Lee, T.H., Harris, C.J., 1998. Adaptive neural network control of robotic manipulators. volume 19. World Scientific.

[25] Lewis, F.L., Dawson, D.M., Abdallah, C.T., 2003. Robot manipulator control: theory and practice. CRC Press.

[26] Lewis, F.L., Vrabie, D., 2009. Reinforcement learning and adaptive dynamic programming for feedback control. IEEE circuits and systems magazine 9, 32–50.

[27] Lewis, F.L., Vrabie, D., Syrmos, V.L., 2012. Optimal control. John Wiley & Sons.

[28] Li, C., Ding, J., Lewis, F.L., Chai, T., 2021a. A novel adaptive dynamic programming based on tracking error for nonlinear discrete-time systems. Automatica 129, 109687.

[29] Li, Y., Chen, L., Tee, K.P., Li, Q., 2015. Reinforcement learning control for coordinated manipulation of multi-robots. Neurocomputing 170, 168–175.

[30] Li, Y., Zhang, J., Liu, W., Tong, S., 2021b. Observer-based adaptive optimized control for stochastic nonlinear systems with input and state constraints. IEEE Transactions on Neural Networks and Learning Systems .

[31] Liu, D., Xue, S., Zhao, B., Luo, B., Wei, Q., 2020. Adaptive dynamic programming for control: A survey and recent advances. IEEE Transactions on Systems, Man, and Cybernetics: Systems 51, 142–160.

[32] Liu, J., 2013. Radial Basis Function (RBF) neural network control for mechanical systems: design, analysis and Matlab simulation. Springer Science & Business Media.

[33] Murray, J.J., Cox, C.J., Lendaris, G.G., Saeks, R., 2002. Adaptive dynamic programming. IEEE transactions on systems, man, and cybernetics, Part C (Applications and Reviews) 32, 140–153.

[34] Naidu, D.S., 2002. Optimal control systems. CRC press.

[35] Nohooji, H.R., 2020. Constrained neural adaptive pid control for robot manipulators. Journal of the Franklin Institute 357, 3907–3923.

[36] Nohooji, H.R., Howard, I., Cui, L., 2018. Neural impedance adaption for assistive human–robot interaction. Neurocomputing 290, 50–59. URL: https://linkinghub.elsevier.com/retrieve/pii/S0925231218301607, doi:10.1016/j.neucom.2018.02.025.

[37] Nohooji, H.R., Howard, I., Cui, L., 2020. Optimal robot-environment interaction using inverse differential riccati equation. Asian Journal of Control 22, 1401–1410. URL: http://doi.wiley.com/10.1002/asjc.2066, doi:10.1002/asjc.2066.

[38] Ouyang, Y., Dong, L., Wei, Y., Sun, C., 2020. Neural network based tracking control for an elastic joint robot with input constraint via actor-critic design. Neurocomputing 409, 286–295.

[39] Ouyang, Y., Sun, C., Dong, L., 2022. Actor–critic learning based coordinated control for a dual-arm robot with prescribed performance and unknown backlash-like hysteresis. ISA transactions 126, 1–13.

[40] Perrusquía, A., Yu, W., 2021. Identification and optimal control of nonlinear systems using recurrent neural networks and reinforcement learning: An overview. Neurocomputing 438, 145–154.

[41] Pham, T.L., Dao, P.N., et al., 2022. Disturbance observer-based adaptive reinforcement learning for perturbed uncertain surface vessels. ISA transactions .

[42] Rahimi Nohooji, H., Howard, I., Cui, L., 2020. Optimal robot-environment interaction using inverse differential riccati equation. Asian Journal of Control 22, 1401–1410.

[43] Rao, A.V., 2009. A survey of numerical methods for optimal control. Advances in the Astronautical Sciences 135, 497–528.

[44] Sanner, R.M., Slotine, J.J.E., 1991. Gaussian networks for direct adaptive control, in: 1991 American control conference, IEEE. pp. 2153–2159.

[45] Saraswat, R., Suhag, S., 2023. Type-2 fuzzy logic pid control for efficient power balance in an ac microgrid. Sustainable Energy Technologies and Assessments 56, 103048.

[46] Shuprajhaa, T., Sujit, S.K., Srinivasan, K., 2022. Reinforcement learning based adaptive pid controller design for control of linear/nonlinear unstable processes. Applied Soft Computing 128, 109450.

[47] Slotine, J.J.E., Li, W., et al., 1991. Applied nonlinear control. volume 199. Prentice hall Englewood Cliffs, NJ.

[48] Song, Y., Huang, X., Wen, C., 2017. Robust adaptive fault-tolerant pid control of mimo nonlinear systems with unknown control direction. IEEE Transactions on Industrial Electronics 64, 4876–4884.

[49] Sutton, R.S., Barto, A.G., 2018. Reinforcement learning: An introduction. MIT press.

[50] Vamvoudakis, K.G., Lewis, F.L., 2010. Online actor-critic algorithm to solve the continuous-time infinite horizon optimal control problem. Automatica 46, 878–888.

[51] Viljamaa, P., Koivo, H.N., 1995. Fuzzy logic in pid gain scheduling, in: Third European Congress on Fuzzy and Intelligent Technologies EUFIT'95, pp. 927–931.

[52] Visioli, A., 2001. Tuning of pid controllers with fuzzy logic. IEE Proceedings-Control Theory and Applications 148, 1–8.

[53] Vrabie, D., Lewis, F.L., 2008. Adaptive optimal control algorithm for continuous-time nonlinear systems based on policy iteration, in: 2008 47th IEEE Conference on Decision and Control, IEEE. pp. 73–79.

[54] Vrabie, D., Lewis, F.L., 2009. Generalized policy iteration for continuous-time systems, in: 2009 International Joint Conference on Neural Networks, IEEE. pp. 3224–3231.

[55] Vrabie, D., Pastravanu, O., Abu-Khalaf, M., Lewis, F.L., 2009. Adaptive optimal control for continuous-time linear systems based on policy iteration. Automatica 45, 477–484.

[56] Wang, C., Hill, D.J., Ge, S.S., Chen, G., 2006. An iss-modular approach for adaptive neural control of pure-feedback systems. Automatica 42, 723–731.

[57] Wang, T., Gao, J., Xie, O., 2022. Sliding mode disturbance observer and q learning-based bilateral control for underwater teleoperation systems. Applied Soft Computing 130, 109684.

[58] Wen, G., Chen, C.P., Feng, J., Zhou, N., 2017. Optimized multi-agent formation control based on an identifier–actor–critic reinforcement learning algorithm. IEEE Transactions on Fuzzy Systems 26, 2719–2731.

[59] Wen, G., Chen, C.P., Ge, S.S., Yang, H., Liu, X., 2019. Optimized adaptive nonlinear tracking control using actor–critic reinforcement learning strategy. IEEE transactions on industrial informatics 15, 4969–4977.

[60] Wen, G., Li, B., Niu, B., 2022. Optimized backstepping control using reinforcement learning of observer-critic-actor architecture based on fuzzy system for a class of nonlinear strict-feedback systems. IEEE Transactions on Fuzzy Systems .

[61] Wiering, M.A., Van Otterlo, M., 2012. Reinforcement learning. Adaptation, learning, and optimization 12, 729.

[62] Yan, L., Liu, Z., Chen, C.P., Zhang, Y., Wu, Z., 2022. Reinforcement learning based adaptive optimal control for constrained nonlinear system via a novel state-dependent transformation. ISA transactions .

[63] Yang, X., He, H., Wei, Q., Luo, B., 2018. Reinforcement learning for robust adaptive control of partially unknown nonlinear systems subject to unmatched uncertainties. Information Sciences 463, 307–322.

[64] Young, W.H., 1912. On the multiplication of successions of fourier constants. Proceedings of the Royal Society of London. Series A, Containing Papers of a Mathematical and Physical Character 87, 331–339.

[65] Yu, H., Xie, T., Paszczyñski, S., Wilamowski, B.M., 2011. Advantages of radial basis function networks for dynamic system design. IEEE Transactions on Industrial Electronics 58, 5438–5450.

[66] Zhang, H., Wei, Q., Liu, D., 2011. An iterative adaptive dynamic programming method for solving a class of nonlinear zero-sum differential games. Automatica 47, 207–214.

[67] Zhang, J., Zhuang, J., Du, H., et al., 2009. Self-organizing genetic algorithm based tuning of pid controllers. Information Sciences 179, 1007–1018.

[68] Zhou, Z.G., Zhou, D., Chen, X., Shi, X.N., 2022. Adaptive actor-critic learning-based robust appointed-time attitude tracking control for uncertain rigid spacecrafts with performance and input constraints. Advances in Space Research .