BỘ GIÁO DỤC VÀ ĐÀO TẠO
TRƯỜNG ĐẠI HỌC BÁCH KHOA HÀ NỘI

CÔNG TRÌNH THAM DỰ
GIẢI THƯỞNG "SINH VIÊN NGHIÊN CỨU KHOA HỌC" CẤP TRƯỜNG
NĂM 2020-2021

Tên công trình:

ADAPTIVE REINFORCEMENT LEARNING OF NONLINEAR SYSTEMS WITH DISTURBANCES BASED ON TIME-VARYING RISE METHOD

Mã số:

Họ và tên sinh viên	Lớp, khóa	Giới tính	Khoa/Viện	Tel
Lê Công Nhật Anh	CTTN - ĐKTĐ K62	Nam	Điện	0942013433
Nguyễn Xuân Khải	CTTN - ĐKTĐ K62	Nam	Điện	0394491201

Giáo viên hướng dẫn: TS. Đào Phương Nam

MÃ ĐỀ TÀI

ADAPTIVE REINFORCEMENT LEARNING OF NONLINEAR SYSTEMS WITH DISTURBANCES BASED ON TIME-VARYING RISE METHOD

Sinh viên: **Lê Công Nhật Anh –** CTTN ĐKTĐ – K62
Nguyễn Xuân Khải – CTTN ĐKTĐ – K62
Giáo viên hướng dẫn: **TS. Đào Phương Nam**
Viện Điện

ABSTRACT

In this paper, a new class of RISE-based adaptive reinforcement learning (ARL) control strategy has been proposed for second-order nonlinear MIMO systems. RISE algorithm has the ability to learn the unknown model uncertainties and external disturbances. RISE control law with sliding variables guarantees tracking performance under restricted assumptions on the uncertainties and nonlinearities of the system. It is proposed to replace some of the static feedback gains in the original RISE control law by nonlinear ones depending on the system states. The concept is based on the fact that nonlinear time-varying gains improve overall efficiency by compensating for a variety of nonlinearities and additive disturbances. In addition, adaptive reinforcement learning is employed to obtain optimal tracking performance for the uncertain/disturbed nonlinear robot system. Thanks to the online actor-critic ADP algorithm based on neural network, the solution of HJB equation was achieved by interation process. All explicit variables and functions in this work contribute to the numerical comparison between two approaches. The obtained simulation results on a 2-DOF robot manipulator show clearly better performance of the proposed time-varying feedback RISE -based ARL control strategy compared to the original controller in terms of tracking accuracy and efficiency.

Keywords: robust optimal control, adaptive reinforcement learning, time-varying RISE, sliding mode control, disturbance.

I. INTRODUCTION

In control theorem and control engineering, optimal control is a significant research area [1,2]. It is concerned with determining a control strategy that optimally steers the dynamics system to equilibrium in terms of a performance index function [3,4]. For linear systems, one must solve the Riccati equation which requires full knowledge of the system dynamics [5,6]. However, under the mathematical viewpoint, finding an optimal controller is equivalent to solving the nonlinear partial differential equation Hamilton-Jacobi-Bellman (HJB) equation, which is difficult to obtain a global analytic solution. In order to obtaining the approximate solution of the HJB equation, several techniques have been proposed, which includes reinforcement learning (RL) [7]. RL is a method for solving optimization problems. It involves an actor or agent that interacts with the environment and modifies its actions or control policies, based on stimuli received in response to its actions [8]. One of the most popular control object for optimality methods is nonlinear dynamical systems.

The motion of a physical systems group such as robotic manipulators, ship, surface vessels, quad-rotor can be considered as mechanical systems with dynamic uncertainties, external disturbances [9]. Furthermore, the actuator saturation and full-state constraint, finite time control have been mentioned in [10] - [15]. Dealing with unknown parameters and disturbances, the terminal sliding mode control (SMC) is one of the remarkable solutions with the consideration of finite-time convergence. In [16], the non-singular terminal sliding surface was employed to obtain the adaptive terminal SMC for a manipulator system. Disturbances including external disturbances, unmodelled dynamics and parameter perturbations, widely exist in aerospace engineering, such as aircrafts, missiles, satellites and also many other engineering systems [17,18]. Generally speaking, disturbance attenuation, noise and time delays are important aspects in control system design [19]. In [20], the multiplicative stochastic link noises were taken into consideration, and distributed filtering problem was successfully solved. In [21], the dissipative control problem for nonlinear Markovian jump systems subject to actuator failures and mixed time-delays were addressed. It is well known that H∞ control is one of the design methods for handling the disturbance attenuation problem of control systems. However, H∞ control is in general too conservative to obtain a highly accurate control performance when treating a modeled disturbance as an unmodeled one [22]. The disturbance observer-based technique was first presented in [23] for a motion servo system. Now, disturbance observer-based control schemes for linear and non-linear systems have been successfully developed and applied in various engineering systems. Later, some disturbance-observer-based control approaches were developed to cope with nonlinear systems in the time domain formulations [18,19,24]. However, nonlinear systems subjected to unknown time-varying external disturbance will further increase the difficulty and complexity for the optimal control system design in practice [25]. Time-varying external disturbance of the nonlinear system needs to be efficiently handled to achieve satisfactory closed-loop control performance.

A novel control mechanism called Robust Integral of the Sign of the Error (RISE) has been developed in [26]. RISE feedback law is a continuous control solution dealing with Multi-Input-Multi-Output (MIMO) high-order nonlinear systems. This non-model-based control strategy can guarantee a semi-global asymptotic tracking under limited assumptions on the system uncertainties and time-varying parameters. RISE -based controllers have been applied in different real-time applications thanks to the robustness and disturbances rejection provided by RISE feedback closed-loop architecture. Because of the powerful robustness and performance acquired by RISE and RISE -based control strategies, the idea of improving such controller arises. Enriching this control law with more nonlinearity may allow it to

compensate for more percentage of high nonlinearities abundant extensively in most of the industrial robotized applications.

According to our investigation, few studies are about the optimal control problems of nonlinear systems with completely unknown disturbance based on ADP [27]. It is known that for improving the disturbance compensation ability, some approaches can be employed to facilitate the direct adaptive control for the uncertain nonlinear system [28]. Thanks to the neural network approximation technique, authors in [29] proposed the novel online ADP algorithm which enables to tune simultaneously both actor and critic terms. The training problem of critic neural network (NN) was determined by modified Levenberg-Marquardt technique to minimize the square residual error. Furthermore, the weights convergence and convergence problem were shown by the weights in actor and critic NN tuning the need of persistence of excitation (PE) condition [29]. Considering the approximate Bellman error, the proposed algorithm in [29] enables to online simultaneously adjust with unknown drift term. Extending this work, by using the special cost function, a model-free adaptive reinforcement learning has been presented without any information of the system dynamics [30]. Furthermore, by integrating the additional identifier, the nonlinear systems were controlled by online adaptive reinforcement learning with completely unknown dynamics [31], [32]. However, these three above works have not mentioned for robotic systems as well as non-autonomous systems yet [30], [31], [32]. In the work of [33], under the consideration of approximation and discrete time systems, online ADP tracking control was proposed for the dynamic of mobile robots. Inspired by the above works and analysis from traditional nonlinear control technique to optimal control strategy, the work focus on the frame of online adaptive reinforcement learning for manipulators and nonlinear control with main contribution are described in the following:

1) Compared to the previous works [9]-[12], [16]-[24], which discussed classical nonlinear controllers in manipulator control systems,, an adaptive reinforcement learning (ARL) -based optimal control design is proposed for an uncertain manipulator system with disturbances in this paper. In comparison to the proposed optimal control in [9], which uses the Kim-Lewis formula in a special case of cost function, ARL-based optimal control architecture has the advantage of being able to handle general performance index for non-autonomous systems with appropriate transform.

2) In contrast to the reinforcement learning scheme -based optimal control in [16], [30]-[33] where mathematical systems of a first-order continuous-time nonlinear autonomous system without any external disturbance are considered, the adaptive dynamic programming, in conjunction with the sliding variable and the time-varying Robust Integral of the Sign of Error (RISE), was used for second-order uncertain/disturbed manipulators in the situation of trajectory tracking control non-autonomous systems.

3) Moreover, this work clarifies initial conditions of the robot manipulator system and presents exploratory signal function. In order to demonstrate the effectiveness of the proposed time-varying feedback RISE -based ARL controller, both the original RISE and the proposed control algorithms were implemented on the 2-DOF robot model. A comparative study between the two implemented controllers is introduced in the simulation section.

The rest of this paper is organized as follows. Section 2.1 presents basic concepts on reinforcement learning. In section 2.2, a background on the original RISE controller is presented. In section 2.3, the proposed contribution to RISE control is introduced. Section 2.4

is dedicated to introduce a combined solution of time-varying RISE -based ARL control schem. In section 2.5, the simulation results and comparison are shown and discussed.

II. RESEARCH OUTPUT

2.1 REINFORCEMENT LEARNING AND OPTIMAL CONTROL

Reinforcement Learning (RL) refers to the problem of a goal-directed agent interacting with an uncertain environment. The goal of an RL agent is to maximize a long-term scalar reward by sensing the state of the environment and taking actions which affect the state. At each step, an RL system gets evaluative feedback about the performance of its action, allowing it to improve the performance of subsequent actions. Several RL methods have been developed and successfully applied in machine learning to learn optimal policies for finite-state finite-action discrete-time Markov Decision Processes (MDPs), shown in Figure 1. An analogous RL control system is shown in Figure 2, where the controller, based on state feedback and reinforcement feedback about its previous action, calculates the next control which should lead to an improved performance. The reinforcement signal is the output of a performance evaluator function, which is typically a function of the state and the control. An RL system has a similar objective to an optimal controller which aims to optimize a long-term performance criterion while maintaining stability. This chapter discusses the key elements in the field of RL and how they can be applied to solve control problems.

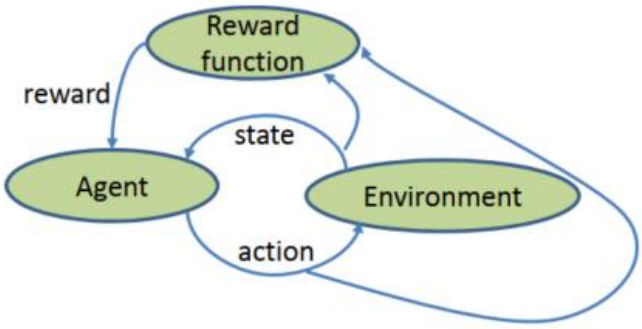


Figure 1 Reinforcement Learning for MDP

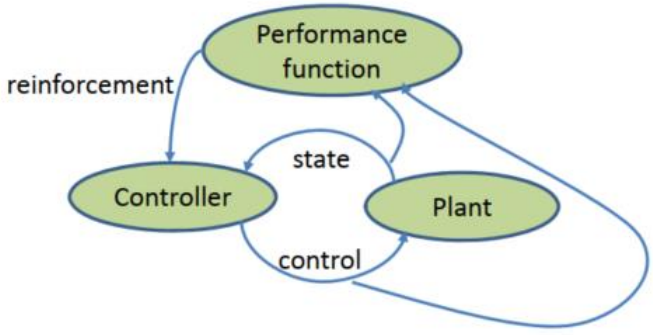


Figure 2 Reinforcement Learning control system

2.1.1 Reinforcement Learning Methods

RL methods typically estimate the value function, which is a measure of goodness of a given action for a given state. The value function represents the reward/penalty accumulated by the agent in the long run, and for a deterministic MDP, may be defined as an infinite-horizon discounted return as [34]

$$V^u\left(x_0\right)=\sum_{k=0}^{\infty}\gamma^k r_{k+1}, \tag{1}$$

where x_k and u_k are the state and action, respectively, for the discrete-time system $x_{k+1}=f\left(x_k,u_k\right)$, $r_{k+1}\triangleq r\left(x_k,u_k\right)$ is the reward/penalty at the k^{th} step, and $\gamma\in\left[0,1\right)$ is the discount factor used to discount future rewards. The objective of an RL method is to determine a policy which maximizes the value function. Since the value function is unknown, typically the first step is to estimate the value function, which can be expressed using Bellman's equation as [34]

$$V^u(x)=r(x,u)+\gamma V^u(f(x,u)) \tag{2}$$

where the index k is suppressed. The optimal value function is defined as

$$V^*(x)=\min_u V^u(x) \tag{3}$$

which can also be expressed using the Bellman optimality condition as

$$\begin{aligned} V^*(x)&=\min_u\left[r(x,u)+\gamma V^*(f(x,u))\right]\\ u^*(x)&=\arg\min_u\left[r(x,u)+\gamma V^*(f(x,u))\right]. \end{aligned} \tag{4}$$

The above Bellman relations form the basis of all RL methods – policy iteration, value iteration, and Q-learning [34, 35, 36]. RL methods can be categorized as model-based and model-free. Model-based or DP-based RL algorithms utilize the expression in (4) but are offline and require perfect knowledge of the environment, as seen from (4). On the other hand, model-free RL algorithms are based on temporal difference (TD), which refers to the difference between temporally successive estimates of the same quantity. In contrast to DP-based RL methods, TD-based RL methods are online and do not use an explicit model of the system, rather they use data (set of samples, trajectories etc.) obtained from the process, i.e., they learn by interacting with the environment. Some of the popular RL methods are subsequently discussed.

a) Policy Iteration

Policy Iteration (PI) algorithms [37, 38] successively alternate between policy evaluation and policy improvement. The algorithm starts with an initial admissible policy, estimates the value function (policy evaluation), and then improves the policy using a greedy search on the estimated value function (policy improvement). The policy evaluation step in DP-based PI is performed using the following recurrence relations until convergence to the value function

$$V^u(x)\leftarrow r(x,u)+\gamma V^u(f(x,u)), \tag{5}$$

where the symbol '$\leftarrow$' denotes the value on the right being assigned to the quantity on the left. After the convergence of policy evaluation, policy improvement is performed using

$$\bar{u}(x)=\arg\min_a\left[r(x,a)+\gamma V^u(f(x,a))\right] \tag{6}$$

It can be seen from (5) and (6) that the DP-based PI algorithm requires knowledge of the system model $f(x,u)$. Using the model-free $TD(0)$ algorithm [39], which learns from

interacting with the environment, this limitation is overcome. Using the $TD(0)$ algorithm, the value function is estimated using the following update

$$V^u(x) \leftarrow V^u(x) + \alpha\left[r(x,u) + \gamma V^u(\bar{x}) - V^u(x) \right] \tag{7}$$

where $\alpha \in (0,1]$ is the learning rate, and $\bar{x}$ denotes the next state observed after performing action u at x. In contrast to DP-based policy evaluation, the value function estimation in (7) does not require an explicit model of the system. The PI algorithm converges to the optimal policy [38]. Online PI algorithms do not wait for the convergence of the policy evaluation step to implement policy improvement; however, their convergence can only be guaranteed only under very restrictive conditions, such as generation of infinitely long trajectories for each iteration [40].

b) Value Iteration

Value Iteration (VI) algorithms directly estimate the optimal value function, which is then used to compute the optimal policy. It combines the truncated policy evaluation and policy improvement steps in one step using the following recurrence relations from DP [34]

$$V(x) \leftarrow \min_a \left[r(x,a) + \gamma V(f(x,a)) \right] \tag{8}$$

VI converges to the optimal $V^*(x)$, and is said to be less computationally intensive than PI, although PI typically converges in fewer iterations [41].

c) Q-Learning

Q-Learning algorithms use Q-factors $Q(x,u)$, which are state-action pairs instead of the state value function $V(x)$. The Q-iteration algorithm uses TD learning to find the optimal Q-factor $Q^*(x,u)$ as

$$Q(x,u) \leftarrow Q(x,u) + \alpha\left[r(x,u) + \gamma \min_a Q(\bar{x},a) - Q(x,u) \right]. \tag{9}$$

The Q-learning algorithm [35] is one of the major breakthroughs in reinforcement learning, since it involves learning the optimal action-value function independent of the policy beingfollowed (also called off-policy), which greatly simplifies the convergence analysis of the algorithm. Adequate exploration is, however, needed for the convergence to Q^*. The optimal policy can be directly found from performing a greedy search on Q^* as

$$u^*(x) = \arg\min_a Q^*(x,a). \tag{10}$$

2.1.2 Aspects of Reinforcement Learning Methods

This section discusses aspects and issues in implementation of the RL methods on high dimensional and large-scale practical systems.

a) Curse of Dimensionality and Function Approximation

RL methods where value function estimates are represented as a table require, at every iteration, storage and updating of all the table entries corresponding to the entire state space. In fact, the computation and storage requirements increase exponentially with the size of the state space, also called the curse of dimensionality. The problem is compounded when considering continuous spaces which contain infinitely many states and actions. One solution approach is to represent value functions using function approximators, which are based on supervised learning, and generalize based on limited information about the state space [34]. A

convenient way to represent value functions is by using linearly parameterized approximators of the form $\theta^T \phi(x)$, where θ is the unknown parameter vector, and ϕ is a user-defined basis function. Selecting the right basis function which represents all the independent features of the value function is crucial in solving the RL problem. Some prior knowledge regarding the process is typically included in the basis function. The parameter vector is estimated using optimization algorithms, e.g., gradient descent, least squares etc. Multi-layer neural networks may also be used as nonlinearly parameterized approximators; however, weight convergence is harder to prove as compared to linearly parameterized network structures.

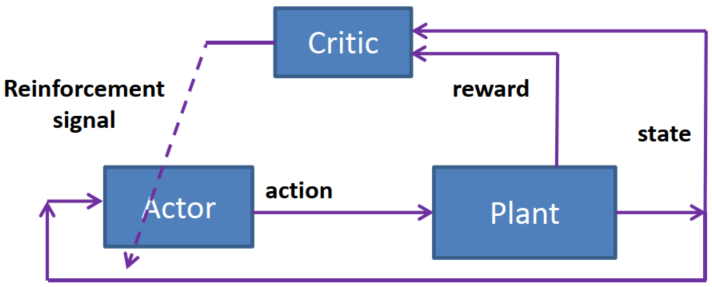


Figure 3 Actor-critic architecture for online policy iteration

b) Actor-Critic Architecture

Actor-critic methods, introduced by Barto [42], implement the policy iteration algorithm online, where the critic is typically a neural network which implements policy evaluation and approximates the value function, whereas the actor is another neural network which approximates the control. The critic evaluates the performance of the actor using a scalar reward from the environment and generates a TD error. The actor-critic neural networks, shown in Figure 3 are updated using gradient update laws based on the TD error.

c) Exploitation Vs Exploration

The trade-off between exploitation and exploration has been a topic of much research in the RL community. For an agent in an unknown environment, exploration is required to try out different actions and learn based on trial and error, whereas past experience may also be exploited to select the best actions and minimize the cost of learning. For sample or trajectory based RL methods (e.g., Monte Carlo) in large dimensional spaces, selecting best actions (e.g., greedy policy) based on current estimates is not sufficient because better alternative actions may potentially never be explored. Sufficient exploration is essential to learn the global optimal solution. However, too much exploration can also be costly in terms of performance and stability when the method is implemented online. One approach is to use a ε-greedy policy, where the exploration is the highest when the agent starts learning, but gradually decays as experience is gained and exploitation is preferred to reach the optimal solution.

2.1.3 Infinite Horizon Optimal Control Problem

RL has close connections with optimal control. In this section, the undiscounted infinite horizon optimal control problem is formulated for continuous-time nonlinear systems. Consider a continuous-time nonlinear system

$$\dot{x} = F(x,u), \tag{11}$$

where $x(t) \in \mathcal{X} \subseteq \mathbb{R}^n, u(t) \in \mathcal{U} \subseteq \mathbb{R}^m$ is the control input, $F : \mathcal{X} \times \mathcal{U} \to \mathbb{R}^n$ is Lipschitz continuous on $\mathcal{X} \times \mathcal{U}$ containing the origin, such that the solution $x(t)$ of the system in (11) is unique for any finite initial condition x_0 and control $u \in \mathcal{U}$. It is also assumed that $F(0,0) = 0$. Further, the system is stabilizable, i.e. there exists a continuous feedback control law $u(x(t))$ such that the closed-loop system is asymptotically stable.

The infinite-horizon scalar cost function for the system (11) can be defined as

$$J(x(t), u(\tau)) = \int_t^{\infty} r(x(s), u(s)) ds \tag{12}$$

where t is the initial time, $r(x,u) \in \mathbb{R}$ is the immediate or local cost for the state and control, defined as

$$r(x,u) = Q(x) + u^T R u, \tag{13}$$

where $Q(x) \in \mathbb{R}$ is continuously differentiable and positive definite, and $R \in \mathbb{R}^{m \times m}$ is a positive-definite symmetric matrix. The optimal control problem is to find an admissible control $u^* \in \Psi(\mathcal{X})$, such that the cost in (12) associated with the system (11) is minimized [43]. An admissible control input $u(t)$ can be defined as a continuous feedback control law $u(x(t)) \in \Psi(\mathcal{X})$, where $\Psi(\cdot)$ denotes the set of admissible controls, which asymptotically stabilizes the system (11) on $\mathcal{X}$, $u(0) = 0$, and $J(\cdot)$ in (12) is finite.

The optimal value function can be defined as

$$V^*(x(t)) = \min_{\substack{u(\tau) \in \Psi(\mathcal{X}) \\ t \le \tau < \infty}} \int_t^{\infty} r(x(s), u(x(s))) ds. \tag{14}$$

Assuming the value function is continuously differentiable, Bellman's principle of optimality can be used to derive the following optimality condition [43]

$$0 = \min_{u(t) \in \Psi(\mathcal{X})} \left[r(x,u) + \frac{\partial V^*(x)}{\partial x} F(x,u) \right] \tag{15}$$

which is a nonlinear partial differential equation (PDE), also called the HJB equation. Based on the assumption that $V^*(x)$ is continuously differentiable, the HJB in (15) provides a means to obtain the optimal control $u^*(x)$ in feedback form. Using the convex local cost in (13) and(15), a closed-form expression for the optimal control can be derived as

$$u^*(x) = -\frac{1}{2} R^{-1} \frac{\partial F(x,u)^T}{\partial u} \frac{\partial V^*(x)^T}{\partial x}. \tag{16}$$

For the control-affine dynamics of the form

$$\dot{x} = f(x) + g(x)u, \tag{17}$$

where $f(x) \in \mathbb{R}^n$ and $g(x) \in \mathbb{R}^{n \times m}$, the expression in (16) can be written in terms of the system state as

$$u^*(x) = -\frac{1}{2} R^{-1} g^T(x) \frac{\partial V^*(x)^T}{\partial x} \tag{18}$$

In general, the solutions to the optimal control problem may not be smooth. Existence of a unique non-smooth solution (called the viscosity solution) is studied.

The HJB in (15) can be rewritten in terms of the optimal value function by substituting for the local cost in (13), the system in (17) and the optimal control in (18), as

$$0 = Q(x) + \frac{\partial V^*(x)}{\partial x} f(x) - \frac{1}{4} \frac{\partial V^*(x)}{\partial x} g(x) R^{-1} g^T(x) \frac{\partial V^*(x)^T}{\partial x}, \quad (19)$$
$$0 = V^*(0).$$

Although in closed-form, the optimal policy in (18) requires knowledge of the optimal value function $V^*(x)$, the solution of the HJB equation in (19). The HJB equation is problematic to solve in general and may not have an analytical solution.

2.1.4 Optimal Control Methods

Since the solution of the HJB is prohibitively difficult and sometimes even impossible, several alternative methods are investigated in literature. The calculus of variations approach generates a set of a first-order necessary optimality conditions, called the Euler-Lagrange equations, resulting in a two-point (or multi-point) boundary value problem, which is typically solved numerically using indirect methods, such as shooting, multiple shooting etc [43]. Another numerical approach is to use direct methods where the state and/or control are approximated using function approximators or discretized using collocation and the optimal control problem is transcribed to a nonlinear programming problem, which can solved using methods such as direct shooting, direct collocation, pseudo-spectral methods etc. Although these numerical approaches are effective and practical, they are open-loop, offline, require exact model knowledge and are dependent on initial conditions. Another approach based on feedback linearization involves robustly canceling the system nonlinearities, thereby reducing the system to a linear system, and solving the associated Algebraic Riccati Equation (ARE)/Differential Riccati Equation (DRE) for optimal control. A drawback of feedback linearization is that it solves a transformed optimal control problem with respect to a part of the control while the other part is used to cancel the nonlinear terms. Moreover, linearization cancels all nonlinearities, some of which may be useful for the system. Inverse optimal controllers circumvent the task of solving the HJB by proving optimality of a control law for a meaningful cost function. The fact that the cost function cannot be chosen a priori by the user limits the applicability of the method.

Given the limitations of methods that seek an exact optimal solution, the focus of some literature has shifted towards developing methods which yield a sub-optimal or an approximately optimal solution. Model-predictive control (MPC) or receding horizon control (RHC) is an example of an online model-based approximate optimal control method which solve the optimal control problem over a finite time horizon at every state transition leading to a state feedback optimal control solution. These methods have been successfully applied in process control where the model is exactly known and the dynamics are slowly varying. An offline successive approximation method improves the performance of an initial stabilizing control by approximating the solution to the generalized HJB (GHJB) equation and then using the Bellman's optimality principle to compute an improved control law. This process is repeated and proven to converge to the optimal policy. The GHJB, unlike the HJB, is a linear PDE which is more tractable to solve, e.g., using methods like the Galerkin projection. The successive approximation method is similar to the policy iteration algorithm in RL; however, the method is offline and requires complete model knowledge. To alleviate the curse of dimensionality associated with dynamic programming, a family of methods, called AC designs (also called ADP) to solve the optimal control problem using RL and neural network

backpropagation algorithms. The methods are, however, applicable only for discrete-time systems and lack a rigorous Lyapunov stability analysis.

2.1.5 Adaptive Optimal Control and Reinforcement Learning

Most optimal control approaches discussed in Section 2.1.4 are offline and require complete model knowledge. Even for linear systems, where the LQR gives the closed-form analytical solution to the optimal control problem, the ARE is solved offline and requires exact knowledge of the system dynamics. Adaptive control provides an inroad to design controllers which can adapt online to the uncertainties in system dynamics, based on minimization of the output error (e.g., using gradient or least squares methods). However, classical adaptive control methods do not maximize a long-term performance function, and hence are not optimal. Adaptive optimal control refers to methods which learn the optimal solution online for uncertain systems. RL methods described in Section 2.1.1 have been successfully used in MDPs to learn optimal polices in uncertain environments, e.g., TD-based Q-learning is an online model-free RL method for learning optimal policies. Sutton et al. argue that RL is a direct adaptive optimal control technique. Owing to the discrete nature of RL algorithms, many methods have been proposed for adaptive optimal control of discrete-time systems. Unfortunately, an RL formulation for continuous-time systems is not as straightforward as in the discrete-time case, because while the TD error in the latter is model-free, it is not the case with the former, where the TD error formulation inherently requires complete knowledge of the system dynamics (15). RL methods based on the model-based TD error for continuous-time systems are proposed. A partial model-free solution is proposed using an actor-critic architecture, however, the resulting controller is hybrid with a continuous-time actor and a discrete-time critic. Other issues concerning RL-based controllers are: closed-loop stability, convergence to the optimal control, function approximation, and tradeoff between exploitation and exploration. Few results have rigorously addressed these issues which are critical for successful implementation of RL methods for feedback control. The work in this dissertation is motivated by the need to provide a theoretical foundation for RL-based control methods and explore their potential as adaptive optimal control methods.

2.2 RISE CONTROL FOR NONLINEAR SYSTEMS

2.2.1 Background on RISE control

First, we examine a first-order, single-input nonlinear system having the general form:

$$m(\eta)\dot{\eta} + f(\eta) = u \tag{20}$$

where $\eta(t) \in \mathbb{R}$ is the system state, $u(t) \in \mathbb{R}$ is the control input, and $m(\eta)$, $f(\eta) \in \mathbb{R}$ are uncertain nonlinear function. It is assumed that $m(\eta)$ and $f(\eta)$ satisfy the following assumptions:

Assumption 1: The function $m(\eta)$ is positive and bounded as follows:

$$\underline{m} \le m(\eta) \le \bar{m}(\eta) \tag{21}$$

where $\underline{m} \in \mathbb{R}$ denotes a positive constant, and $\bar{m}(\eta) \in \mathbb{R}$ denotes a positive non-decreasing function.

Assumption 2: The functions $m(\eta)$ and $f(\eta)$ are second-order differentiable with respect to $\eta(t)$ such that

$$\begin{aligned} &m(\eta), \frac{\partial m(\eta)}{\partial \eta}, \frac{\partial^2 m(\eta)}{\partial \eta^2} \in \mathcal{L}_\infty \quad \text{if} \quad \eta(t) \in \mathcal{L}_\infty \\ &f(\eta), \frac{\partial f(\eta)}{\partial \eta}, \frac{\partial^2 f(\eta)}{\partial \eta^2} \in \mathcal{L}_\infty \quad \text{if} \quad \eta(t) \in \mathcal{L}_\infty \end{aligned} \tag{22}$$

Let $\eta_d(t) \in \mathbb{R}$ be a given reference trajectory that is continuously differentiable up to its third derivative such that

$$\frac{d^i \eta_d(t)}{dt^i} \in \mathcal{L}_\infty, \ i = 0,1,2,3 \tag{23}$$

To quantify the control objective, we define the tracking error $e(t) \in \mathbb{R}$ as follows

$$e \triangleq \eta_d - \eta \tag{24}$$

Our objective is to obtain asymptotic tracking with a continuous control law using (23) and norm-based, inequality bounds on the functions $\frac{\partial^i m(\eta_d)}{\partial \eta_d^i}$ and $\frac{\partial^i f(\eta_d)}{\partial \eta_d^i}$, $i = 0,1,2$.

Remark 1: For simplicity of presentation, we have assumed $m(\eta)$ and $f(\eta)$ do not depend explicitly on time or on unknown time-varying parameters. However, it should be emphasized that the proposed control approach can compensate for these phenomena provided the time-varying effects satisfy second-order differentiability conditions similar to those given in (22). That is, the functions $m(\eta)$ and $f(\eta)$ could be easily replaced by $m\left(\eta, \theta_1(t), t\right)$ and $f\left(\eta, \theta_2(t), t\right)$ where $\theta_i(t), \ i = 1,2$ denote unknown time-varying parameter vectors and other time-varying disturbance that may appear nonlinearly in the model.

RISE control law that can achieve the control objective is generally defined as follows:

$$u(t) = \left(k_s + 1\right) e(t) - \left(k_s + 1\right) e\left(t_0\right) + \int_{t_0}^{t} \left[\left(k_s + 1\right) \alpha e(\tau) + \beta \operatorname{sgn}(e(\tau))\right] d\tau \tag{25}$$

where $k_s, \alpha, \beta \in \mathbb{R}$ are positive control gains, t_0 is the initial time, and $\operatorname{sgn}(\cdot)$ denotes the standard signum function. The control law of (25) ensures asymptotic tracking provided the control gains k_s and β are chosen sufficiently large relative to the norm of the initial tracking error and a reference trajectory-based bound, respectively

$$k_s > \frac{1}{4\lambda_3} \rho_0^2 \left(\sqrt{\frac{\lambda_2\left(\eta(t_0)\right)}{\lambda_1}} \eta_0 \right) \tag{26}$$

$$\beta > \left|N_d(t)\right| + \frac{1}{\alpha} \left|\dot{N}_d(t)\right| \tag{27}$$

Based on the stability analysis introduced in [26], the signal (25) has the ability to learn the unknown system model.

2.2.2 Time-varying RISE

The original controller in (25) can be split up into two components: a linear feedback part based on the measured combined error e, and a nonlinear signum function. The linear

part consists of proportional and integral actions on the combined error, which is similar to a PI controller but taking as input the combined error instead of the position error. These two linear control actions may lead up to poor performances when dealing with highly nonlinear systems at critical dynamic operating conditions. They have considerable sensitivity to disturbances and limited tuning abilities.

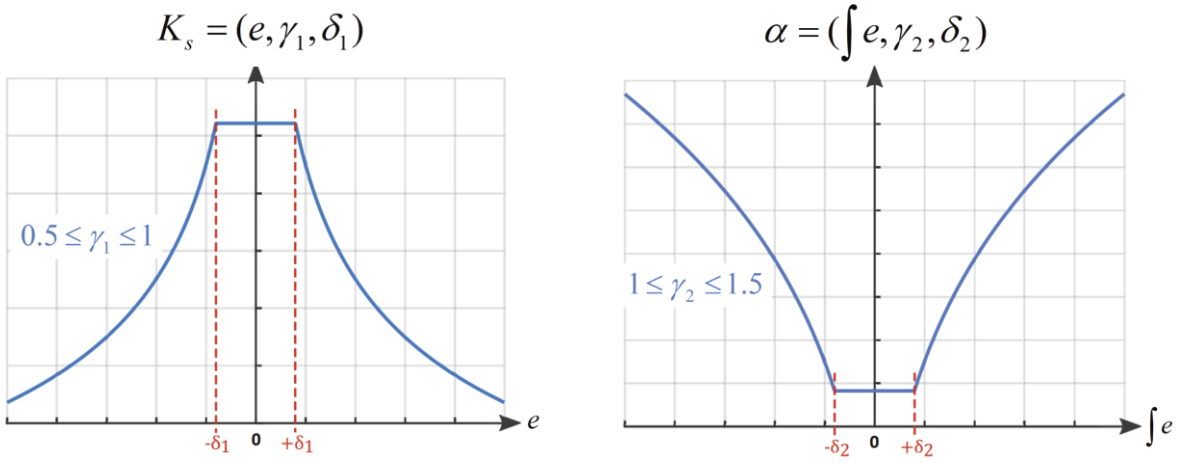
It is proposed to replace the proportional and the integral static feedback gains by nonlinear time-varying ones. The proposed time-varying feedback RISE controller is given as follows:

$$u(t) = \left(K_s(.)+1\right)e(t) - \left(K_s(t_0)+1\right)e(t_0) + \int_{t_0}^{t}\left[\left(k_{s0}+1\right)\alpha(.)e(\tau) + \beta\,\mathrm{sgn}(e(\tau))\right]d\tau \qquad (28)$$

with $K_s(.)$ and $\alpha(.)$ are two nonlinear feedback functions designed in:

$$K_s(.) \equiv K_s(e,\gamma_1,\delta_1) = \begin{cases} k_{s0}\left|e\right|^{\gamma_1-1}, & \left|e\right| > \delta_1 \\ k_{s0}\delta_1^{\gamma_1-1}, & \left|e\right| \le \delta_1 \end{cases} \qquad (29)$$

$$\alpha(.) \equiv \alpha(e,\gamma_2,\delta_2) = \begin{cases} \alpha_0\left|\int e\right|^{\gamma_2-1}, & \left|\int e\right| > \delta_2 \\ \alpha_0\delta_2^{\gamma_2-1}, & \left|\int e\right| \le \delta_2 \end{cases} \qquad (30)$$



(a) Evolution of $K_s(.)$ (b) Evolution of $\alpha(.)$

Figure 4 Evolution of the proposed nonlinear gains with respect to their arguments

where $k_{s0}, \alpha_0, \gamma_1, \delta_1, \gamma_2, \delta_2$ are positive design parameters need to be chosen carefully. Indeed, to meet the desired performance, γ_1 and γ_2 are chosen within the intervals $[0.5,1]$ and $[1,1.5]$ respectively.

On one hand, the selection of γ_1 within the interval $[0.5,1]$ can reduce the proportional gain $K_s(.)$ at high combined error values and increase it at small ones (see Figure 4-a). As long as the combined error remains within the small interval $[-\delta_1,\delta_1]$ around zero, the proportional gain remains constant as a maximum saturated value. Notice that the combined error gives knowledge about both position and velocity errors. Thus, such variation

of the gain could result in a rapid transition of the closed-loop system states and a favorable damping.

On the other hand, the nonlinear feedback gain $\alpha(.)$ varies as function of the integral of the combined error (see Figure 4-b), which means that it is more concerned with the steady state combined errors (i.e., errors that persist with time). The choice of $\alpha(.)$ within the interval $[1,1.5]$ gives large integral gain for the large steady state combined errors, and small integral gain for the small steady state combined errors as illustrated in Figure 4-b. As long as this error remains within the small interval $[-\delta_2, \delta_2]$ around zero, the integral gain remains as a minimum constant value. This variation may accelerate the tracking process towards the set point and prevent the integral term from accumulating above or below specific bounds which can solve the integral windup problem.

Choosing γ_1 and γ_2 in their corresponding intervals leads to globally bounded nonlinear functions as follows (bounds can be realized from Figure 4):

$$0 < K_{sm} \triangleq k_{s0} \|e\|_\infty^{\gamma_1 - 1} \le K_s(.) \le k_{s0} \delta_1^{c_1 - 1} \triangleq K_{sM} \tag{31}$$

$$0 < \alpha_{2m} \triangleq \alpha_{20} \delta_2^{\gamma_2 - 1} \le \alpha_2(.) \le \alpha_{20} \left\| \int e \right\|_\infty^{\lambda_2 - 1} \triangleq \alpha_{2M} \tag{32}$$

where $\|.\|_\infty$ indicates the infinity-norm.

Including the above-mentioned time-varying feedback gains in the standard equation of a RISE controller may boost the controller's global tracking efficiency and robustness to changes in system parameters. It's worth double-checking that the nonlinear function structure is easy enough to incorporate in real-time experiments.

2.3 PROPOSED APPROACH: TIME-VARYING RISE FOR ADAPTIVE REINFORCEMENT LEARNING OF NONLINEAR SYSTEMS

2.3.1 Robot Manipulator Model

Consider the planar robot manipulator systems described by the following dynamic equation:

$$M(\eta)\ddot{\eta} + C(\eta, \dot{\eta})\dot{\eta} + G(\eta) + F(\dot{\eta}) + d(t) = \tau(t) \tag{33}$$

where $M(\eta) \in \mathbb{R}^{n \times n}$ is a generalized inertia matrix, $C(\eta, \dot{\eta}) \in \mathbb{R}^{n \times n}$ is a generalized centripetal-Coriolis matrix, $G(\eta) \in \mathbb{R}^n$ is a gravity vector, $F(\dot{\eta}) \in \mathbb{R}^n$ is a generalized friction, $d(t)$ is a vector of disturbances, $\tau(t)$ is the vector of control inputs. It is worth emphasizing that the above manipulator belongs to the class of Euler-Lagrange systems, which has the following special property [12]:

Property 1: The inertia symmetric matrix $M(\eta)$ is positive definite, and satisfies $\forall \xi \in \mathbb{R}^n$:

$$\underline{m}\|\xi\|^2 \le \xi^T M(\eta)\xi \le \bar{m}(\eta)\|\xi\|^2 \tag{34}$$

$$\xi^T (\dot{M}(\eta) - 2C(\eta, \dot{\eta}))\xi = 0 \tag{35}$$

where $\underline{m} \in \mathbb{R}$ is a positive constant, $\bar{m}(\eta) \in \mathbb{R}$ is a positive non-decreasing function with respect to η. Notice that $\|\cdot\|$ stands for the classical Euclidean norm.

Several following assumptions will be employed in considering the stability later.

Assumption 3: If $\eta(t), \dot{\eta}(t) \in L_\infty$, then all these functions $C(\eta, \dot{\eta})$, $F(\dot{\eta})$, $G(\eta)$ and the first, second partial derivatives of all functions of $M(\eta)$, $C(\eta, \dot{\eta})$, $G(\eta)$ with respect to $\eta(t)$ as well as of the elements of $C(\eta, \dot{\eta})$, $F(\dot{\eta})$ with respect to $\dot{\eta}(t)$ exist and are bounded.

Assumption 4: The desired trajectory $\eta_d(t)$ as well as the first, second, third and fourth time derivatives of it exist and are bounded.

Assumption 5: The vector of external disturbance term $d(t)$ and the derivatives with respect to time of $d(t)$ are bounded by known constants.

The control objective is to ensure the system tracks a desired time-varying trajectory $n_{ref}(t)$ in presence of dynamic uncertainties by using the frame of online adaptive reinforcement learning based optimal control design and disturbance attenuation technique. Considering the sliding variable $s(t) = \dot{e}_1 + \alpha_1 e_1$ ($\alpha_1 \in \mathbb{R}^{n \times n} > 0$, $e_1(t) = \eta_{ref} - \eta$) and the coresponding sliding surface as follows:

$$M = \{e_1(t) \in \mathbb{R}^n : s(t) = 0\} \tag{36}$$

According to (1), the dynamic equation of the sliding variable $s(t)$ can be given as:

$$M\dot{s} = -Cs - \tau + f + d \tag{37}$$

where $f(\eta, \dot{\eta}, \eta_{ref}, \dot{\eta}_{ref}, \ddot{\eta}_{ref})$ is nonlinear function defined:

$$f = M(\ddot{\eta}_{ref} + \alpha_1 \dot{e}_1) + C(\dot{\eta}_{ref} + \alpha_1 e_1) + G + F \tag{38}$$

Remark 2: The role of above sliding variable is considered to reduce the order of second-order uncertain/disturbed manipulator systems. It enable us to employ the adaptive reinforcement learning for a first-order continuous-time nonlinear autonomous system. Additionally, the external disturbance $d(t)$ and nonlinear function f are handled by time-varying RISE in the next section.

2.3.2 Control Desgin

Assume that the dynamic model of robot manipulator is known, the control input can be designed as:

$$\tau = f + d - u \tag{39}$$

where the term u is designed by using optimal control algorithm and the remaining term $f + d$ will be estimated later. Therefore, it can be seen that:

$$M\dot{s} = -Cs + u \tag{40}$$

According to (36) and (40), we obtain the following time-varying model:

$$\dot{x} = \begin{bmatrix} -\alpha_1 e_1 + s \\ -M(\eta_{ref} - e_1)^{-1} C(\eta_{ref} - e_1, \dot{\eta}_{ref} + \alpha_1 e_1 - s)s \end{bmatrix} + \begin{bmatrix} 0_{n \times n} \\ M^{-1} \end{bmatrix} u \tag{41}$$

where $x = [e_1^T, s^T]^T$ and the infinite horizon cost function to be minimized is

$$J(x,u) = \int_0^\infty (\frac{1}{2} x^T Q x + \frac{1}{2} u^T R u) dt \tag{42}$$

where $Q \in \mathbb{R}^{2n\times 2n}$ and $R \in \mathbb{R}^{n\times n}$ are positive definite symmetric matrices.

However, in order to deal with the problem of tracking control, some additional states are given. This work leads us to avoid the non-autonomous systems. Subsequently, the adaptive reinforcement learning is considered to find optimal control solution for autonomous affine state-space model with the assumption that the desired trajectory $\eta_{ref}(t)$ satisfies $\dot{\eta}_{ref}(t) = f_{ref}(\eta_{ref})$

$$\dot{X} = A(X) + B(X)u \tag{43}$$

where $X = [x^T, \eta_{ref}^T, \dot{\eta}_{ref}^T]^T$ (44)

$$A(X) = \begin{bmatrix} -\alpha_1 e_1 + s \\ -M(\eta_{ref} - e_1)^{-1} C(\eta_{ref} - e_1, \dot{\eta}_{ref} + \alpha_1 e_1 - s)s \\ \eta_{ref} \\ f_{ref}(\eta_{ref}) \end{bmatrix} \tag{45}$$

$$B(X) = \begin{bmatrix} 0_{n\times n} \\ M^{-1} \\ 0_{2n\times n} \end{bmatrix} \tag{46}$$

Define the new infinite horizon integral cost function to be minimized is

$$J(X,u) = \int_t^{\infty} (\frac{1}{2} X^T Q_T X + \frac{1}{2} u^T R u) d\tau \tag{47}$$

where

$$Q_T = \begin{bmatrix} Q & 0 \\ 0 & 0 \end{bmatrix} \tag{48}$$

In order to guarantee the stability in optimal control design, we can consider the class of "Admissible Policy" described in [29], [30]:

Definition 1 [29], [30] (Admissible Policy): A control input $\mu(X)$ is call as admissible in term of (47) on U, if $\mu(X)$ is continuous on U and the affine system (43) was stabilized by this control signal $\mu(X)$ on U and $J(X)$ is finite for any $X \in U$.

The optimal control objective can now be considered finding an admissible control signal $\mu^*(X)$ such that the cost function (47) associated with affine system (47) is minimized. According to the classical Haminlton-Jacobi-Bellman (HJB) equation theory [32], the optimal controller $u^*(X)$ and equivalent optimal cost function $V^*(X)$ are derived as:

$$u^*(X) = -\frac{1}{2} R^{-1} B^T(X) \frac{\partial V^*(X)^T}{\partial X} \tag{49}$$

$$H^*\left(X, u^*, \frac{\partial V^*}{\partial X}\right) = \frac{\partial V^*}{\partial X}\left(A(X) + B(X)u^*\right) + \frac{1}{2} X^T Q_T X + \frac{1}{2} u^{*T} R u^* = 0 \tag{50}$$

However, it is hard to directly solve the HJB equation as well as offline solution requires complete knowledge of the mathematical model. Thus, the simultaneous learning

based online solution is considered by using Neural Networks to represent the optimal cost function and the equivalent optimal controller [32]:

$$V(X) = W^T \psi(X) + \varepsilon_v(X) \tag{51}$$

$$u^*(X) = -\frac{1}{2} R^{-1} B^T(X) \left(\left(\frac{\partial \psi}{\partial X} \right)^T W + \left(\frac{\partial \varepsilon_v(X)}{\partial X} \right)^T \right) \tag{52}$$

where $W \in \mathbb{R}^N$ is vector of unknown ideal NN weights, N is the number of neurons, $\psi(X) \in \mathbb{R}^N$ is a smooth NN activation function, $\varepsilon_v(X) \in \mathbb{R}$ is the function reconstruction error. The objective of establishing the NN (51) is to find the actor/critic NN updating laws $\hat{W}_a$, $\hat{W}_c$ to approximate the actor and critic parts obtaining the optimal control law without solving the HJB equation (more details see [32]). Moreover, the smooth NN activation function is chosen depending on the description of manipulators (see Section 2.5). In [32], the Weierstrass approximation theorem enables us to uniformly approximate not only $V^*(X)$ but also $\frac{\partial V^*(X)}{\partial X}$ with $\varepsilon_v(X)$, $\frac{\partial \varepsilon_v(X)}{\partial X} \to 0$ as $N \to \infty$. Consider to fix the number N, the critic $\hat{V}(X)$ and the actor $\hat{u}(X)$ are employed to approximate the optimal cost function and the optimal controller as:

$$\hat{V}(X) = \hat{W}_c^T \psi(X) \tag{53}$$

$$\hat{u}(X) = -\frac{1}{2} R^{-1} B^T(X) \left(\frac{\partial \psi}{\partial X} \right)^T \hat{W}_a \tag{54}$$

The adaptation laws of critic $\hat{W}_c$ and actor $\hat{W}_a$ weights are simultaneously implemented to minimize the integral squared Bellman error and the squared Bellman error δ_{hjb}, respectively.

$$\delta_{hjb} = \hat{H}\left(X, \hat{u}, \frac{\partial \hat{V}}{\partial X} \right) - H^*\left(X, u^*, \frac{\partial V^*}{\partial X} \right) = \hat{W}_c^T \sigma + \frac{1}{2} X^T Q_T X + \frac{1}{2} \hat{u}^T R \hat{u} \tag{55}$$

where $\sigma(X, \hat{u}) = \frac{\partial \psi}{\partial X}(A + B\hat{u})$ is the critic regression vector. Similar to the work in [32], the adaptation law of Critic weights is given:

$$\frac{d}{dt} \hat{W}_c = -k_c \lambda \frac{\sigma}{1 + \upsilon \sigma^T \lambda \sigma} \delta_{hjb} \tag{56}$$

where υ, k_c are constant positive gains, and $\lambda \in \mathbb{R}^{N \times N}$ is a symmetric estimated gain matrix computed as follows:

$$\frac{d}{dt} \lambda = -k_c \lambda \frac{\lambda \sigma^T}{1 + \upsilon \sigma^T \psi \sigma} \lambda; \qquad \lambda(t_s^+) = \lambda(0) = \varphi_0 I \tag{57}$$

where t_s^+ is resetting time satisfying $\alpha_{\min}\{\lambda(t)\} \le \varphi_1$, $\varphi_0 > \varphi_1$. This work ensures $\lambda(t)$ is positive definite and prevents the covariance wind-up problem [32].

$$\varphi_1 I \le \lambda(t) \le \varphi_0 I \tag{58}$$

The actor adaptation law can be described as:

$$\frac{d}{dt}\hat{W}_a = -\frac{k_{a1}}{\sqrt{1+\sigma^T\sigma}}\frac{\partial\psi}{\partial X}BR^{-1}B^T\frac{\partial\psi^T}{\partial X}\left(\hat{W}_a-\hat{W}_c\right)\delta_{hjb}-k_{a2}\left(\hat{W}_a-\hat{W}_c\right) \tag{59}$$

It is necessary to guarantee of PE conditions of the critic regression vector in using this developed algorithm. Unlike linear systems, where PE conditions of the regression translates to sufficient richness of the external input, there is no verifiable method exists to ensure PE regression translates in nonlinear regulation problems. To ensure PE qualitatively, an exploratory signal $n(t)$ consisting of sinusoids of varying frequencies is added to the control in the first time of learning process.

Remark 3: The approximate/adaptive reinforcement learning (ARL) control law (Actor) and approximately optimal cost function (Critic) are obtained in (54) and (53), respectively. Based on the optimization principle, the updated law of Actor and Critic are carried out as in (59) and (56). Compared with the optimal control law in [1], the ARL control algorithm has the advantage in that it is able to handle for general performance index. The convergences of estimated actor/critic weights $\hat{W}_a$ and $\hat{W}_c$ depend on the PE condition of $\frac{\sigma}{\sqrt{1+\upsilon\sigma^T\lambda\sigma}}\in\mathbb{R}^N$ in [32]. Unlike the work in [32], this algorithm do not mentioned the identifier design and focuses on the manipulator control design. Moreover, the learning technique in adaptation law (59) and (56) is different from data-driven online integral reinforcement learning in [29], [30]. It is worth noting that a clear functionalized exploratory signal as well as clear initial conditions of the system is described in this work instead of random variables, which clarifies the learning process and contributes to the comparison of different approaches. These will be described in Section 2.5. In order to develop this adaptive reinforcement learning for manipulator systems in the trajectory tracking control problem, it is necessary to consider the manipulator dynamic as affine systems (43).

Thus, the control design (39) is finalized by integrating the estimation of $\varepsilon = f + d$, which is designed based on the time-varying RISE framework [1].

The proposed time-varying RISE structure is presented as in Section 2.2.2

$$\varepsilon(t) = (K_s(.)+1)s(t) - (K_s(t_0)+1)s(0) + \rho(t) \tag{60}$$

$$\frac{d}{dt}\rho = (k_{s0}+1)\alpha(.)s(t) + \beta\,\mathrm{sgn}(s(t)) \tag{61}$$

In summary, the control input is described as

$$\tau = \varepsilon - u + n \tag{62}$$

Remark 4: In early works [9], the optimal control design was considered for uncertain/disturbed mechanical systems by the RISE framework. The tracking control objective of this optimal control law is satisfied by appropriate assumptions 3-5 [9]. However, it should be noted that the work in [9] is extended by integrating adaptive reinforcement learning in the trajectory tracking problem with the consideration of non-autonomous systems, which are not directly applied the adaptive reinforcement learning. The proposed control scheme in [9] only considered the optimal control in the special case of cost function, which leads to the optimal control problem was easily implemented by using the formula of Kim and Lewis [9] for this special case. However, it is worth emphasizing that the method of Kim and Lewis in [9] is not able to carry out for general case. Compared with the proposed

controller in [1], RISE based uncertainties/disturbance estimation has the advantage in that it is able to combine with adaptive reinforcement learning algorithm for HJB equation to deal with general performance index. Moreover, this work deals with optimal control problem (41) for the general performance index (42) required the appropriate algorithm being adaptive reinforcement learning (ARL) for HJB equation. Additionally, due to the non-autonomous property of model (41), it is not able to directly carry out the model (41) by ARL strategy. Therefore, we proposed the transform method to obtain the modified autonomous system (11) developed by ARL algorithm. On the other hand, it should be noted that authors in [32] considered an online ARL-based method for a first-order continuous-time nonlinear autonomous system without any external disturbance. However, unlike the work in [32], a disturbed manipulator is described by a second-order continuous-time nonlinear systems (33). Therefore, in order to employ ARL strategy, the sliding variable is proposed in this work to reduce the order of manipulator model.

Remark 5: Moreover, this paper improve the standard RISE framework by implementing time-varying nonlinear functions, which generalizes the control problems. Including the above-mentioned time-varying feedback gains in the standard equation of a RISE controller may boost the controller's global tracking efficiency and robustness to changes in system parameters. It is also important that the nonlinear functions' structure is easy enough to incorporate in real-time experiments.

2.5 SIMULATION RESULTS

2.5.1 Simulation Setup

This section describes the evaluation of the performance of the proposed controllers through simulation tests. Both the original RISE and the proposed time-varying RISE methods with ARL were implemented on the two-link robot manipulator. A comperison between the two employed controllers is studied in the next sessions.

A 2-DOF planar robot manipulator system, which is modeled by Euler-Lagrange formulas (33). In the case of 2-DOF planar robot manipulator systems $(n=2)$, the above matrices in (33) can be represented as follows:

$$
\begin{aligned}
M(\eta) &= \begin{bmatrix} \zeta_1 + 2\zeta_2 \cos\eta_2 & \zeta_3 + \zeta_2 \cos\eta_2 \\ \zeta_3 + \zeta_2 \cos\eta_2 & \zeta_3 \end{bmatrix}, \\
G(\eta) &= \begin{bmatrix} \zeta_4 \cos\eta_1 + \zeta_5 \cos\left(\eta_1 + \eta_2\right) \\ \zeta_5 \cos\left(\eta_1 + \eta_2\right) \end{bmatrix}, \\
C(\eta, \dot{\eta}) &= \begin{bmatrix} -\zeta_2 \sin\eta_2 \dot{\eta}_2 & -\zeta_2 \sin\eta_2 \left(\dot{\eta}_1 + \dot{\eta}_2\right) \\ \zeta_2 \sin\eta_2 \dot{\eta}_1 & 0 \end{bmatrix}
\end{aligned}
\tag{63}
$$

where $\zeta_i, i = 1...5$ are constant parameters depending on mechanical parameters and gravitational acceleration. In this simulation, these constant parameters are chosen as

$$\zeta_1 = 5, \quad \zeta_2 = 1, \quad \zeta_3 = 1, \quad \zeta_4 = 1.2, \quad \zeta_5 = g. \tag{64}$$

The simulation scenario was considered to validate the performance of proposed controller as follows:

The time-varying desired reference signal is defined as $\eta_{ref} = \begin{bmatrix} 3sin(t) & 3cos(t) \end{bmatrix}^T$ with the vector of disturbances is given as $d(t) = \begin{bmatrix} 50sin(t) & 50cos(t) \end{bmatrix}^T$ For the control objective

of general cost function, the optimal control problem is implemented with the arbitrary positive definite symmetric matrices in cost function (47) as:

$$Q = \begin{bmatrix} 40 & 2 & -4 & 4 \\ 2 & 40 & 4 & -6 \\ -4 & 4 & 4 & 0 \\ 4 & -6 & 0 & 4 \end{bmatrix}, \qquad R = \begin{bmatrix} 0.25 & 0 \\ 0 & 0.25 \end{bmatrix} \tag{65}$$

Moreover, due to the stability description of sliding surface, the design parameters in sliding variable $s(t) = \dot{e}_1 + \alpha_1 e_1$ are chosen to satisfy that $\alpha_1 \in \mathbb{R}^{n\times n}$ is a constant positive definite matrix:

$$\alpha_1 = \begin{bmatrix} 15.6 & 10.6 \\ 10.6 & 10.4 \end{bmatrix} \tag{66}$$

For the purpose of stability of the closed system as well as uncertainties/disturbances estimation, the remaining control gains in original RISE framework are chosen in (25) as:

$$\alpha = \begin{bmatrix} 60 & 0 \\ 0 & 60 \end{bmatrix}, \quad k_s = \begin{bmatrix} 100 & 0 \\ 0 & 100 \end{bmatrix}, \quad \beta = 5 \tag{67}$$

and the time-varying RISE parameters as in (28), (29), and (30)

$$\begin{aligned} &k_{s0} = \begin{bmatrix} 100 & 0 \\ 0 & 100 \end{bmatrix}, \qquad \gamma_1 = 0.9, \qquad \delta_1 = 0.05 \\ &\alpha_0 = \begin{bmatrix} 24 & 0 \\ 0 & 24 \end{bmatrix}, \qquad \gamma_2 = 1.5, \qquad \delta_2 = 1.3 \\ &\beta = 5 \end{aligned} \tag{68}$$

The gains in Actor-Critic learning laws are selected guaranteeing (56)-(59) as

$$k_c = 800, \quad \upsilon = 1, \quad k_{a1} = 0.01, \quad k_{a2} = 1. \tag{69}$$

On the other hand, according to [1], the consideration of V in (51) can be calculated precisely as

$$V = 2x_1^2 - 4x_1x_2 + 3x_2^2 + 2.5x_3^2 + x_3^2\cos(\eta_2) + x_3x_4 + x_3x_4\cos(\eta_2) + 0.5x_4^2 \tag{70}$$

Although we can choose the arbitrary $\psi(X)$ in (51). However, for the comparison between result from experiences and result in (70), it leads to that the $\psi(X)$ was chosen as

$$\psi(X) = \left[x_1^2, x_1x_2, x_2^2, x_3^2, x_3^2\cos(\eta_2), x_3x_4, x_3x_4\cos(\eta_2), x_4^2\right]^T \tag{71}$$

According to (29), exact value of $\hat{W}_c$ in (53) and $\hat{W}_a$ in (54) are

$$W = \begin{bmatrix} 2 & -4 & 3 & 2.5 & 1 & 1 & 1 & 0.5 \end{bmatrix}^T \tag{72}$$

In the simulation, the covariance matrix is initialized as

$$\lambda(0) = \text{diag}\begin{pmatrix} 100 & 300 & 300 & 1 & 1 & 1 & 1 & 1 \end{pmatrix} \tag{73}$$

All the NN weights $\hat{W}_c$, $\hat{W}_a$ are initialized as

$$\hat{W}_c(0) = \begin{bmatrix} 0.6 & 0.1 & 0.7 & 0.5 & 0.5 & 0.5 & 0.7 & 0.6 \end{bmatrix}^T$$
$$\hat{W}_a(0) = \begin{bmatrix} 0.6 & 0.2 & 0.2 & 0.9 & 1 & 1 & 0.4 & 0.2 \end{bmatrix}^T \quad (74)$$

and the states and the its first time derivative are initialized as

$$q(0) = \begin{bmatrix} 0.5 & 0 \end{bmatrix}^T$$
$$\dot{q}(0) = \begin{bmatrix} 0.9 & 0.8 \end{bmatrix}^T \quad (75)$$

To ensure PE qualitatively, an exploratory signal consisting of sinusoids of varying frequencies is added to the control for the first 25 seconds after 35 seconds of simulation time.

$$n(t) = \begin{bmatrix} n_1(t) & n_2(t) \end{bmatrix}^T$$
$$n_1(t) = 40(sin(-25t)^2 cos(35t) + sin(-20t)^2 cos(3t)) \quad (76)$$
$$n_2(t) = 40(sin(26t)^2 cos(29t) + sin(20t)^2 cos(4t))$$

In order to quantify the relevance of the control algorithm, we need to define a certain performance index. One of our main objectives is to enhance the precision and increase the tracking accuracy of robot through the proposed controller. An accuracy evaluation tool frequently used to evaluate differences between a desired trajectory and a measured one is the Root-Mean-Square Error (RMSE) criterion. It can quantify approximately the error between the desired trajectory and the actual one traversed by the robot.

$$RMSE = \sqrt{\frac{1}{N}\sum_{i=1}^{N}\left(e_{1,1}^2(i) + e_{1,2}^2(i)\right)} \quad (77)$$

where $e_{1,1}, e_{1,2}$ denote the joints tracking errors. N is the number of the collected samples through the whole trajectory.

In order to estimate the energy consumption for each controller at high dynamic operating conditions, the input-torques-based criterion is proposed as follows.

$$E_T = \sum_{i=1}^{2}\sum_{j=1}^{N}\left|\tau_i(j)\right| \quad (78)$$

where the control efforts E_T is the total summation of the absolute value of the input torques delivered by the two actuators.

To determine the convergence error of the trainning process, we calculate the differences between trained weights and precise weights.

$$CE = \left|\hat{W} - W\right| \quad (79)$$

The next session will quantitatively and visually demonstrate the simulation results and comparison between the two methods.

2.5.2 Result Analysis

Table 1 Control performance evaluation for both controllers

	Original Optimal RISE	Optimal Time-Varying RISE	Comments

Weights	2.0000 2.0000 -4.0003 -4.0000 3.0002 3.0000 2.5000 2.5000 1.0000 1.0000 1.0000 1.0000 1.0000 1.0000 0.5000 0.5000	2.0339 2.0000 -4.0396 -4.0000 2.9719 3.0000 2.5011 2.5000 1.0007 1.0000 0.9996 1.0000 0.9997 1.0000 0.5002 0.5000	Less precise, acceptable
CE	0.0003	0.0592	
RMSE	0.3214	0.3264	1.56% worse, acceptable
E_T	7.0056e+06	4.4316e+06	36.74% better

Table 1 notes some explicit information to compare the original RISE and the proposed time-varying RISE methods with ARL were implemented on the two-link robot manipulator.

In general, their weight convergences in approximating value function using neural network are all excellent. The weights within the original optimal RISE approach converges precisely to the solution in (72) while the proposed method shows less precise but acceptable final weight values.

Regarding tracking errors RMSE, the original method produces just 1.56% better result than the novel one. Following the reference trajectory shown in Figure 5, the joint tracking errors for both controllers are registered and plotted in Figure 6. Particularly, Figure 5 and Figure 6 explain that time-varying RISE -based ARL controller produces larger overshoot with smaller oscillatory frequency at transcient time. It is worth noting that the proposed method also accelerates settling time of the system and guarantees stable zero tracking errors, which is significantly better than the original RISE -based ARL controller. The total RMSE indexes are nearly similar for both controllers.

Another important note is the great improvement in terms of energy consumption which is a 36.74% reduction with the novel method.

Because of the extended nonlinear feedback gains and their different behavior, the proposed time-varying RISE -based ARL control clearly outperforms the original RISE control in terms of precision and efficiency.

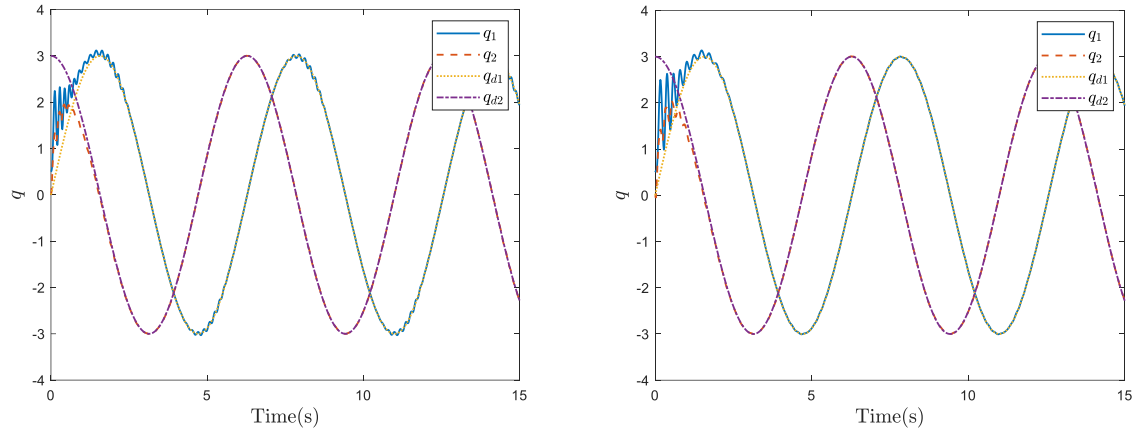


Figure 5 Tracking trajectories of the two controllers

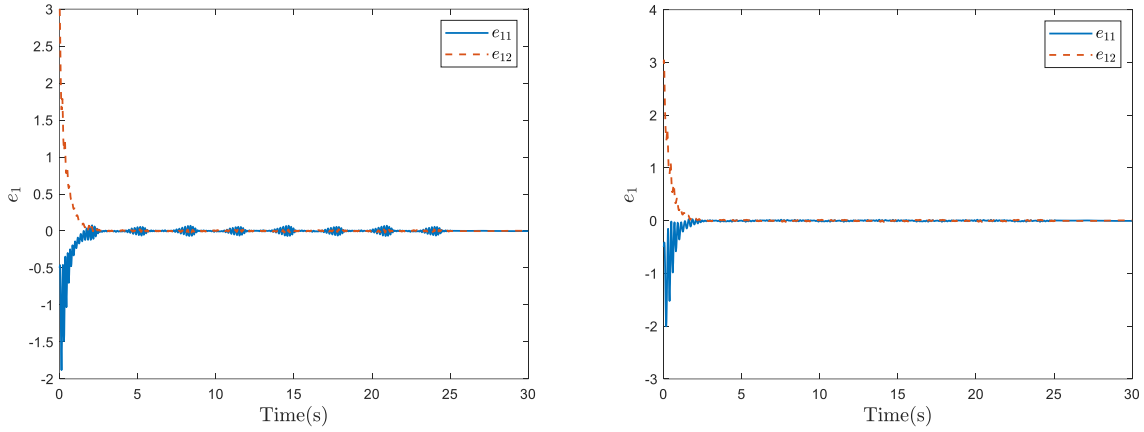


Figure 6 Tracking errors of the two controllers

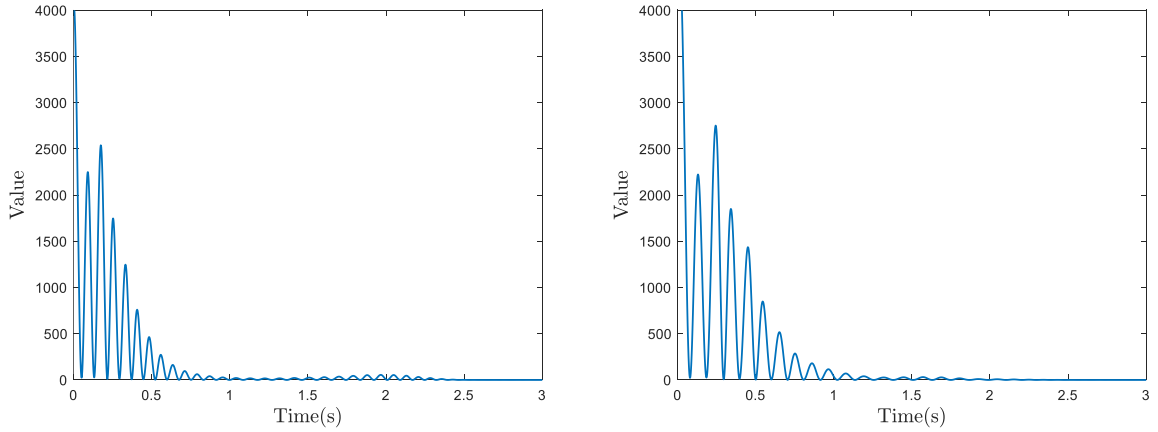


Figure 7 Cost function value of of the two controllers

The proposed time-varying RISE optimal controller results in better cost function value through time, which can be seen from Figure 7. In contrast, there is an unstable rise at 2 second in the cost value produced by the standard strategy.

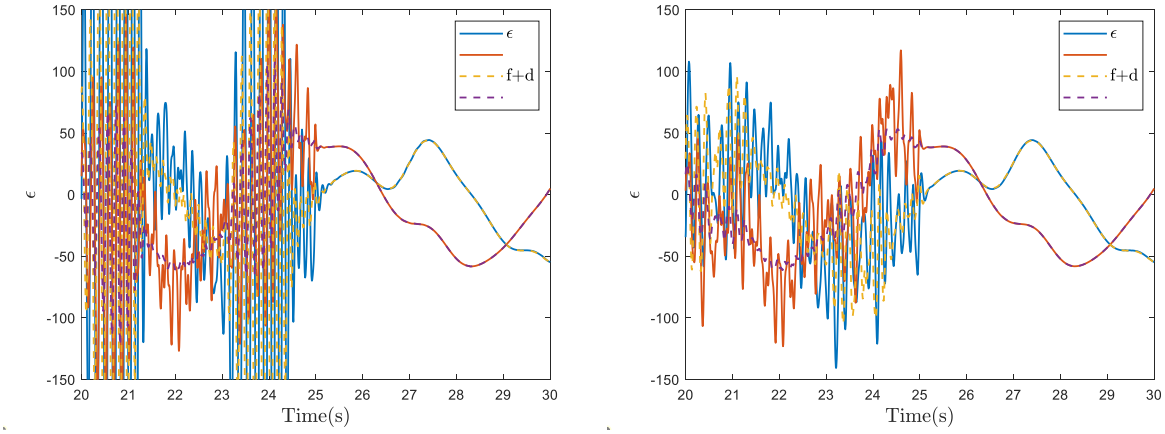


Figure 8 Estimation of the two controllers

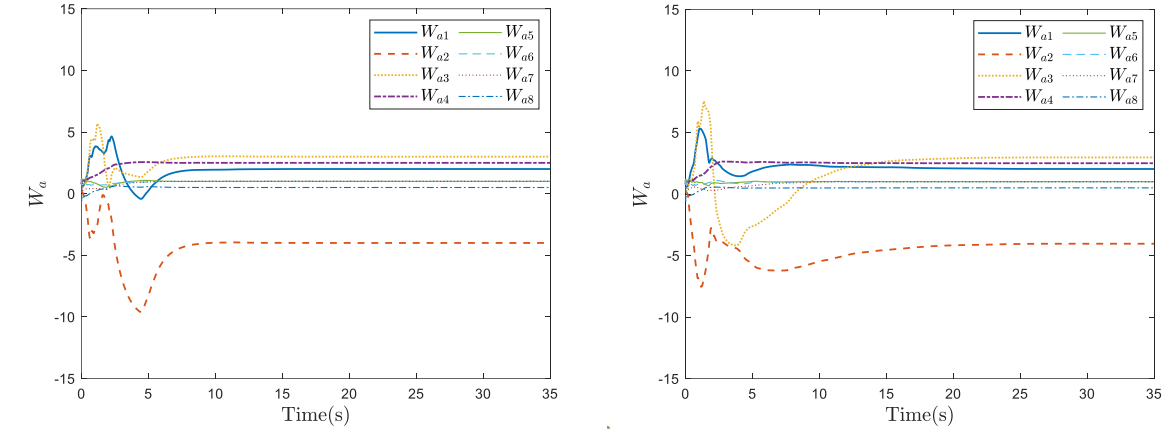


Figure 9 Actor weights of the two controllers

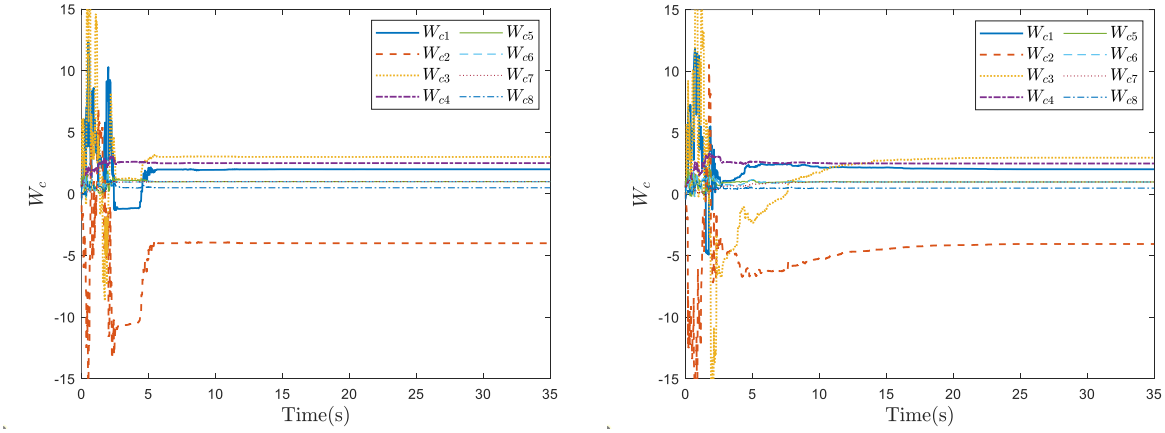


Figure 10 Critic weights of the two controllers

Figure 8 demonstrates the improved estimation performance of novel approach towards uncertainties and disturbances during the exploration and learning process. After that, the estimation of both two methods towards uncertainties and disturbances is brilliant.

The convergence processes of actor and critic weights in training neural network are shown from Figure 9 and Figure 10. While the original controller leads to faster weight convergence, the proposed time-varying RISE -based control method reduces overshoot phenomenon.

III. CONCLUSION

This study addresses a robust optimal control method for a class of uncertain nonlinear systems with unknown disturbances. In this framework, after defining sliding variable, an online adaptive reinforcement learning (ARL) is presented to achieve the optimality. Actor-critic neural networks (NNs) is considered to approximate the Hamilton–Jacobi–Bellman equation. Based on the robust integral of the sign of the error (RISE) method, uncertain/disturbed components of the systems are estimated, which guarantees the trajectory tracking objective. Moreover, this work proposes a new time-varying RISE which is combined with ARL structure in order to obtain improvements. Simulation results on two-link robot manipulator demonstrate the performance of the proposed robust optimal control scheme.

This work can be further extended by considering the completely unknown dynamics of the system in off-policy integral reinforcement learning with the proposed time-varying feedback RISE controller. Moreover, co-operative control of multiple robot manipulators is also an interesting topic where ARL-based control algorithm can be implemented.

IV. REFERENCES

[1] H. Zhang, C. Qing, Y. Luo, Neural-network-based constrained optimal control scheme for discrete-time switched nonlinear system using dual heuristic programming, IEEE Trans. Autom. Sci. Eng. 11 (3) (2014) 839–849.

[2] H. Zhang, C. Qing, B. Jiang, Y. Luo, Online adaptive policy learning algorithm for H∞ state feedback control of unknown affine nonlinear discrete-time systems, IEEE Trans. Cybern. 44 (12) (2014) 2706–2718.

[3] Q. Wei, D. Liu, H. Lin, Value iteration adaptive dynamic programming for optimal control of discrete-time nonlinear systems, IEEE Trans. Cybern. 46 (3) (2016) 840–853.

[4] Q. Wei, D. Liu, X. Yang, Infinite horizon self-learning optimal control of nonaffine discrete-time nonlinear systems, IEEE Trans. Neural Netw. Learn. Syst. 26 (4) (2015) 866–879.

[5] F. Lewis, A general riccati equation solution to the deadbeat control problem, IEEE Trans. Autom. Control 27 (1) (1982) 186–188

[6] Y. Jiang, Z. Jiang, Computational adaptive optimal control for continuous-time linear systems with completely unknown dynamics, Automatica 48 (10) (2012) 2699–2704.

[7] F. Lewis, K. Vamvoudakis, Reinforcement learning for partially observable dynamic processes: adaptive dynamic programming using measured output data, IEEE Trans. Syst. Man Cybern. Part B Cybern. 41 (1) (2011) 14–25.

[8] R. Sutton, A. Barto, Reinforcement learning: an introduction, A Bradford Book, The MIT Press, Cambridge, Massachusetts, London, England, 2005.

[9] Dupree, Keith and Patre, Parag M and Wilcox, Zachary D and Dixon, Warren E. (2011). Asymptotic optimal control of uncertain nonlinear Euler-Lagrange systems. Automatica,1, 99-107

[10] Hu, Xin and Wei, Xinjiang and Zhang, Huifeng and Han, Jian and Liu, Xiuhua. (2019). Robust adaptive tracking control for a class of mechanical systems with unknown disturbances under actuator saturation. International Journal of Robust and Nonlinear Control,6(29), 1893-1908.

[11]. Yang, Liang and Yang, Jianying. (2011). Nonsingular fast terminal sliding-mode control for nonlinear dynamical sys tems. International Journal of Robust and Nonlinear Control,21(16), 1865-1879

[12]. Guo, Yong and Huang, Bing and Li, Ai-jun and Wang, Chang-qing. (2019). Integral sliding mode control for EulerLagrange systems with input saturation. International Journal of Robust and Nonlinear Control,29(4), 1088-1100.

[13]. He, Wei and Chen, Yuhao and Yin, Zhao. (2015). Adaptive Neural Network Control of an Uncertain Robot With FullState Constraints. IEEE transactions on cybernetics,46(3), 620-629.

[14]. He, Wei and Dong, Yiting. (2017). Adaptive fuzzy neural network control for a constrained robot using impedance learning. IEEE transactions on neural networks and learning systems,29(4), 1174-1186.

[15]. He, Wei and Dong, Yiting and Sun, Changyin. (2015). Adaptive neural impedance control of a robotic manipulator with input saturation. IEEE Transactions on Systems, Man, and Cybernetics: Systems,46(3), 334-344

[16]. Mondal, Sanjoy and Mahanta, Chitralekha. (2014). Adaptive second order terminal sliding mode controller for robotic manipulators. Journal of the Franklin Institute,351(4), 2356-2377.

[17] J. Wu, J. Huang, Y. Wang, K. Xing, Nonlinear disturbance observer-based dynamic surface control for trajectory tracking of pneumatic muscle system, IEEE Trans. Control Syst. Technol. 22 (2) (2014) 440–455.

[18] W. Chen, Disturbance observer based control for nonlinear systems, IEEE/ASME Trans. Mechatron. 9 (4) (2004) 706–710.

[19] J. Yang, W. Chen, S. Li, Non-linear disturbance observer-based robust control for systems with mismatched disturbances/uncertainties, IET Control Theory Appl. (2011) 2053–2062.

[20] L. Ma, Z. Wang, Y. Liu, F.E. Alsaadi, Distributed filtering for nonlinear time-delay systems over sensor networks subject to multiplicative link noises and switching topology, Int. J. Robust Nonlinear Control 29 (10) (2019) 1–19.

[21] L. Ma, Z. Wang, Q. Han, Y. Liu, Dissipative control for nonlinear Markovian jump systems with actuator failures and mixed time-delays, Automatica 98 (2018) 358–362

[22] J. Yang, W. Chen, S. Li, Non-linear disturbance observer-based robust control for systems with mismatched disturbances/uncertainties, IET Control Theory Appl. (2011) 2053–2062.

[23] L. Ma, Z. Wang, Y. Liu, F.E. Alsaadi, Distributed filtering for nonlinear time-delay systems over sensor networks subject to multiplicative link noises and switching topology, Int. J. Robust Nonlinear Control 29 (10) (2019) 1–19.

[24] L. Guo, W. Chen, Disturbance attenuation and rejection for systems with nonlinearity via DOBC approach, Int. J. Robust Nonlinear Control 15 (3) (2005) 109–125.

[25] M. Chen, S.S. Ge, Direct adaptive neural control for a class of uncertain nonaffine nonlinear systems based on disturbance observer, IEEE Trans. Cybern. 43 (4) (2013) 1213–1225.

[26] Xian, B., Dawson, D. M., Queiroz, M. S., & Chen, J. (2004, July). A Continuous Asymptotic Tracking Control Strategy for Uncertain Nonlinear Systems. IEEE Transactions on Automatic control, 49(7), 1206-1211

[27] R. Song, F. Lewis, Q. Wei, H. Zhang, Off-policy actor-critic structure for optimal control of unknown systems with disturbances, IEEE Trans. Cybern. 46 (5) (2016) 1041–1050.

[28] S. Mohammed, W. Huo, J. Huang, H. Rifaï, Y. Amirat, Nonlinear disturbance observer based sliding mode control of a human-driven knee joint orthosis, Robot. Auton. Syst. 75 (A) (2016) 41–49.

[29] Vamvoudakis, Kyriakos G and Vrabie, Draguna and Lewis, Frank L. (2014). Online adaptive algorithm for optimal control with integral reinforcement learning. International Journal of Robust and Nonlinear Control,24(17), 2686-2710.

[30] Zhu, Yuanheng and Zhao, Dongbin and Li, Xiangjun. (2016). Using reinforcement learning techniques to solve continuous-time non-linear optimal tracking problem without system dynamics. IET Control Theory & Applications,10(12), 1339-1347.

[31] Lv, Yongfeng and Na, Jing and Yang, Qinmin and Wu, Xing and Guo, Yu. (2016). Online adaptive optimal control for continuous-time nonlinear systems with completely unknown dynamics. International Journal of Control,89(1), 99-112.

[32] Bhasin, Shubhendu and Kamalapurkar, Rushikesh and Johnson, Marcus and Vamvoudakis, Kyriakos G and Lewis, Frank L and Dixon, Warren E. (2013). A novel actor-critic-identifier architecture for approximate optimal control of uncertain nonlinear systems. Automatica,49(1), 82-92.

[33] Li, Shu and Ding, Liang and Gao, Haibo and Liu, YanJun and Huang, Lan and Deng, Zongquan. (2020). ADPbased online tracking control of partially uncertain timedelayed nonlinear system and application to wheeled mobile robots. IEEE transactions on cybernetics,50(7), 3182-3194.

[34] R. Sutton and A. Barto, Introduction to reinforcement learning. MIT Press Cambridge, MA, USA, 1998.

[35] C. Watkins and P. Dayan, "Q-learning," Mach. Learn., vol. 8, no. 3, pp. 279–292, 1992.

[36] D. White and D. Sofge, Handbook of intelligent control: neural, fuzzy, and adaptive approaches. Van Nostrand Reinhold Company, 1992

[37] R. Bellman, Dynamic Programming. Dover Publications, Inc., 2003

[38] R. Howard, Dynamic programming and Markov processes. Technology Press of Massachusetts Institute of Technology (Cambridge), 1960

[39] R. Sutton, "Learning to predict by the methods of temporal differences," Mach. Learn., vol. 3, no. 1, pp. 9–44, 1988

[40] J. Tsitsiklis, "On the convergence of optimistic policy iteration," The Journal of Machine Learning Research, vol. 3, pp. 59–72, 2003

[41] D. Bertsekas, Dynamic Programming and Optimal Control. Athena Scientific, 2007

[42] A. Barto, R. Sutton, and C. Anderson, "Neuronlike adaptive elements that can solve difficult learning control problems," IEEE Trans. Syst. Man Cybern., vol. 13, no. 5, pp. 834–846, 1983

[43] D. Kirk, Optimal Control Theory: An Introduction. Dover Pubns, 2004.