

Article

Adaptive Optimal Robust Control for Uncertain Nonlinear Systems Using Neural Network Approximation in Policy Iteration

Dengguo Xu ^{1,2} , Qinglin Wang ¹ and Yuan Li ^{1,*}

¹ School of Automation, Beijing Institute of Technology, Beijing 100081, China; xudengguo@bit.edu.cn (D.X.); wangql@bit.edu.cn (Q.W.)

² School of Electrical Engineering, Liupanshui Normal University, Liupanshui 553004, China

* Correspondence: liyuan@bit.edu.cn

Abstract: In this study, based on the policy iteration (PI) in reinforcement learning (RL), an optimal adaptive control approach is established to solve robust control problems of nonlinear systems with internal and input uncertainties. First, the robust control is converted into solving an optimal control containing a nominal or auxiliary system with a predefined performance index. It is demonstrated that the optimal control law enables the considered system globally asymptotically stable for all admissible uncertainties. Second, based on the Bellman optimality principle, the online PI algorithms are proposed to calculate robust controllers for the matched and the mismatched uncertain systems. The approximate structure of the robust control law is obtained by approximating the optimal cost function with neural network in PI algorithms. Finally, in order to illustrate the availability of the proposed algorithm and theoretical results, some numerical examples are provided.

Keywords: policy iteration; uncertain nonlinear system; robust control; adaptive optimal control



Citation: Xu, D.; Wang, Q.; Li, Y. Adaptive Optimal Robust Control for Uncertain Nonlinear Systems Using Neural Network Approximation in Policy Iteration. *Appl. Sci.* **2021**, *11*, 2312. <https://doi.org/10.3390/app11052312>

Academic Editor: Emanuele Carpanzano and Giovanni Petrone

Received: 21 January 2021
Accepted: 28 February 2021
Published: 5 March 2021

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2021 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

It is ineluctable to contain uncertain parameters and disturbances in practical systems due to modeling errors, external disturbances, and so on [1]. Thus, it is of great practical significance to study robust control of the uncertain systems. In recent years, the control problem of uncertain systems has been extensively studied. The literature on robust control of uncertain systems mainly includes linear systems and nonlinear systems. For the uncertain linear systems, algebraic Riccati equations (ARE) or linear matrix inequalities (LMI) were mostly used to deal with them in the classical research methods [2–6]. In this literature, both matched and mismatched systems were involved. For nonlinear systems, the early research methods include feedback linearization, fuzzy modeling, nonlinear H_∞ control, and so on [7–11]. However, in recent decades, neural networks (NN) and PI in RL were used to approximate the robust control law numerically [12–14].

The PI method in RL was initially utilized to calculate the optimal control law for some deterministic systems. Werbos first proposed an idea of approximating the solution in Bellman's equation using approximate dynamic programming (ADP) [15]. There have been many results of using the PI method to calculate the optimal control law for some deterministic systems [16–19]. There are two major benefits of using the PI algorithm to deal with such optimal problems. On the one hand, it can effectively solve some problems of the “curse of dimensionality” in engineering control [20]. On the other hand, it can be utilized to calculate the optimal control law without knowing the system dynamics. In the practice of engineering control, it is difficult to obtain system dynamics accurately. Therefore, it is a good choice to use the PI algorithm to solve unknown model control problems.

Within the last ten years, the PI method was also developed to calculate the robust controller for some uncertain systems, which is based on the optimal control method of

robust control [21]. For an input constraint nonlinear system with continuous time, a novel RL-based algorithm was proposed to deal with the robust control problem in [22]. Based on network structure approximation, an online PI algorithm was developed to solve robust control of a class of nonlinear discrete-time uncertain systems in [23]. Using a data-driven RL algorithm, a robust control scheme was developed for a class of completely unknown dynamic systems with uncertainty in [24]. In addition, there are many other examples of literature on robust control based on RL, such as [25–27]. In all the literature listed above, the solution to the Hamilton Jacobi Bellman (HJB) equation was approximated by neural network. In fact, solving the HJB equation is a key problem in optimal control problem [28]. The HJB equation is difficult to solve because it is a nonlinear partial differential equation. For a nonlinear system, the HJB equation is solved with neural network approximation in many cases. Meanwhile, for the linear system, ARE is used to solve it instead of a neural network. However, for all we know, most of the current studies have not considered the input uncertainty in system. The input uncertainty does exist in the actual control system.

In this study, a class of continuous-time nonlinear systems with internal and input uncertainties is considered. The main objective is to establish robust control laws for the uncertain systems. By solving the optimal control problem constructed, the robust control problem is converted into calculating an optimal controller. The online PI algorithms are proposed to calculate robust control by approximating the optimal cost with neural network. The convergence of the proposed algorithms is proved. Numerical examples are given to illustrate the availability of the method.

Our main contributions are as follows. First, more general uncertain nonlinear systems are considered, in which the uncertainty entered both the system and the input. For the matched and the mismatched uncertain systems, it is proved that the robust control can be converted into calculating an optimal controller. Second, the online PI algorithms are developed to solve the robust control problem. The neural network is utilized to approximate the optimal cost in PI algorithm, which fulfilled a difficult task of solving the HJB equation.

The rest of this paper is arranged as follows. We formulate the robust control problems and propose some basic results for the issues under consideration in Section 2. Solving the robust control problem is converted to calculate an optimal control law of a nominal or auxiliary system in Sections 3 and 4. Based on approximating optimal cost with neural network, the online PI algorithms are developed to solve the robust control problem in Section 5. To support the proposed theoretical framework, we provide some numerical examples in Section 6. In Section 7, the study is concluded, and the scope for future research is discussed.

2. Preliminaries and Problem Formulation

Consider an uncertain nonlinear system as follows:

$$\dot{x}(t) = f(x(t)) + \Delta f(x(t)) + [g(x(t)) + \Delta g(x(t))]u(t), x(0) = x_0 \quad (1)$$

where $x(t) \in \mathbb{R}^n$ is the system state, $u(t) \in \mathbb{R}^m$ is the control input, $f(x(t)) \in \mathbb{R}^n$, $g(x(t)) \in \mathbb{R}^{n \times m}$ are known function, $\Delta f(x(t)) \in \mathbb{R}^n$, $\Delta g(x(t)) \in \mathbb{R}^{n \times m}$ are uncertain disturbing function.

The control objective is to establish a robust control law $u = u(x)$ in order that the closed-loop system is asymptotically stable for all allowed uncertain disturbances $\Delta f(x(t))$ and $\Delta g(x(t))$.

As a general case, we first make the following assumptions to ensure that the state Equation (1) is well defined [1,29].

Assumption 1. In (1), $f(x) + g(x)u$ is Lipschitz continuous with respect to x and u on the set $\Omega \subseteq \mathbb{R}^n$ containing the origin.

Assumption 2. For the free vibration system, $f(0) = 0, \Delta f(0) = 0$, that is, $x = 0$ is the equilibrium point of the free vibration system.

Definition 1. The system (1) is called to satisfy the system dynamics matched condition if there is a function matrix $h(x) \in \mathbb{R}^{m \times 1}$ such that

$$\Delta f(x) = g(x)h(x). \quad (2)$$

Definition 2. System (1) is called to satisfy the input matched condition if there is a function $m(x) \in \mathbb{R}^{m \times m}$ such that

$$\Delta g(x) = g(x)m(x), \quad (3)$$

where $m(x) \geq 0$.

Definition 3. If the system (1) satisfies the conditions (2) and (3) for any allowed disturbances $\Delta f(x)$ and $\Delta g(x)$, then the system (1) is called a matched uncertain system.

Definition 4. If the system (1) does not satisfy the condition (2) or (3) for any allowed disturbances $\Delta f(x)$ and $\Delta g(x)$, then the system (1) is called a mismatched uncertain system.

Next, we consider the robust control problem of nonlinear system (1) with matched and mismatched conditions, respectively.

3. Robust Control of Matched Uncertain Nonlinear Systems

This section considers the problem of robust control when the system (1) meets the matched conditions (2) and (3). By constructing appropriate performance indexes, the problem of robust control is transformed into calculating the optimal control law of a corresponding nominal system. Based on the optimal control of the nominal system, a PI algorithm is proposed to obtain robust feedback controller.

For the nominal system

$$\dot{x} = f(x) + g(x)u, \quad (4)$$

find the controller $u = u(x)$ to minimize performance index

$$J(x_0, u) = \int_0^{\infty} [f_{max}^2(x) + x^T x + u^T u] dt \quad (5)$$

where $f_{max}(x)$ is the supremum function of uncertainty $h(x)$, that is $\|h(x)\| \leq f_{max}(x)$.

The definition of admissible control in optimal control problem is given below [26].

Definition 5. The control policy $u(x)$ is called an admissible control of the system (4) with regard to the performance function (5) on compact set $\Omega \subseteq \mathbb{R}^n$ if $u(x)$ is continuous on Ω , $u(0) = 0$, it can stabilize the system (4) on Ω , and the performance function (5) is limited for any $x \in \Omega$.

According to the performance index (5), the cost function corresponding to the admissible control $u(x)$ is given by

$$V(x, u) = \int_t^{\infty} [f_{max}^2(x) + x^T x + u^T u] dt \quad (6)$$

Taking time derivative on both side of (6), it follows the Bellman equation

$$f_{max}^2(x) + x^T x + u^T u + \nabla V^T [f(x) + g(x)u] = 0, \quad (7)$$

where ∇V is the gradient vector of the cost function $V(x, u)$ with respect to x .

Definite Hamiltonian function

$$H(x, u, \nabla V) = f_{max}^2(x) + x^T x + u^T u + \nabla V^T [f(x) + g(x)u] \quad (8)$$

Determining the extremum of the Hamiltonian function yields the optimal control function

$$u^*(x) = -\frac{1}{2}g^T(x)\nabla V \quad (9)$$

By substituting (9) into (7), it follows that optimal cost $V^*(x)$ satisfies the following HJB equation

$$f_{max}^2(x) + x^T x + \nabla V^{*T} f(x) - \frac{1}{4}\nabla V^{*T} g(x)g^T(x)\nabla V^* = 0 \quad (10)$$

and initial conditions $V^*(0) = 0$.

Solving the optimal cost $V^*(x)$ from the HJB Equation (10), we can get the solution to the optimal control problem. Thus, the robust control problem can be solved.

The following theorem shows that optimal control $u^*(x) = -\frac{1}{2}g^T(x)\nabla V^*$ is a robust controller for matched uncertain systems.

Theorem 1. Assume that the conditions (2) and (3) hold in system (1) and the solution $V^*(x)$ in HJB Equation (10) exists. Considering the nominal nonlinear system (4) with performance index (5), then the optimal control policy $u^*(x) = -\frac{1}{2}g^T(x)\nabla V^*$ can globally asymptotically stabilize the nonlinear uncertain system (1). That is to say, the closed-loop uncertain system $\dot{x}(t) = f(x(t)) + \Delta f(x(t)) + [g(x(t)) + \Delta g(x(t))]u^*(x)$ is globally asymptotically stable.

Proof. In order to prove the stability with controller $u^*(x) = -\frac{1}{2}g^T(x)\nabla V^*$, $V^*(x)$ is chosen as the Lyapunov function. Considering the performance index (5), $V^*(x)$ is obviously positive, and $V^*(0) = 0$. Taking time derivative of the function $V^*(x)$ along closed-loop system (1), it follows that

$$\frac{dV^*}{dt} = \nabla V^{*T}[f(x) + \Delta f(x)] - \frac{1}{2}\nabla V^{*T}[g(x) + \Delta g(x)]g^T(x)\nabla V^* \quad (11)$$

Using the matched conditions (2) and (3), it follows from (11) that

$$\begin{aligned} \frac{dV^*}{dt} = & \nabla V^{*T} f(x) + \nabla V^{*T} g(x)h(x) - \frac{1}{2}\nabla V^{*T} g(x)g^T(x)\nabla V^* \\ & - \frac{1}{2}\nabla V^{*T} g(x)m(x)g^T(x)\nabla V^* \end{aligned} \quad (12)$$

From HJB Equation (10), one can obtain

$$\nabla V^{*T} f(x) = -f_{max}^2(x) - x^T x + \frac{1}{4}\nabla V^{*T} g(x)g^T(x)\nabla V^* \quad (13)$$

Substituting (13) into (12) yields

$$\begin{aligned} \frac{dV^*}{dt} = & -f_{max}^2(x) - x^T x + \frac{1}{4}\nabla V^{*T} g(x)g^T(x)\nabla V^* + \nabla V^{*T} g(x)h(x) - \frac{1}{2}\nabla V^{*T} g(x)g^T(x)\nabla V^* \\ & - \frac{1}{2}\nabla V^{*T} g(x)m(x)g^T(x)\nabla V^* \\ = & -f_{max}^2(x) - x^T x - \frac{1}{4}\nabla V^{*T} g(x)g^T(x)\nabla V^* + \nabla V^{*T} g(x)h(x) - \frac{1}{2}\nabla V^{*T} g(x)m(x)g^T(x)\nabla V^* \end{aligned} \quad (14)$$

It follows from $m(x) \geq 0$ that $-\frac{1}{2}\nabla V^{*T} g(x)m(x)g^T(x)\nabla V^* \leq 0$. Therefore, from (14), we can have

$$\begin{aligned}
 \frac{dV^*}{dt} &\leq -f_{max}^2(x) - x^T x - \frac{1}{4} \nabla V^{*T} g(x) g^T(x) \nabla V^* + \nabla V^{*T} g(x) h(x) \\
 &= -f_{max}^2(x) - x^T x - \frac{1}{4} [\nabla V^{*T} g(x) g^T(x) \nabla V^* - 4 \nabla V^{*T} g(x) h(x) + 4 h(x)^T h(x)] + h(x)^T h(x) \\
 &= -x^T x + h(x)^T h(x) - f_{max}^2(x) - \frac{1}{4} [g^T(x) \nabla V^* - 2h(x)]^T [g^T(x) \nabla V^* - 2h(x)] \\
 &\leq -x^T x
 \end{aligned}
 \tag{15}$$

Therefore, by Lyapunov stability theory [30], the optimal control $u^*(x) = -\frac{1}{2} g^T(x) \nabla V^*$ can make the matched uncertain system (1) asymptotically stable. Thus, for a constant $c > 0$, there is a neighborhood $N = \{x : \|x\| < c\}$ such that if the state $x(t)$ enters the neighborhood N , then $x \rightarrow 0$ when $t \rightarrow \infty$. However, $x(t)$ cannot stay out of the domain N forever; otherwise, for all $t > 0$, there is $\|x(t)\| \geq c$. This implies that

$$\begin{aligned}
 V^*[x(t)] - V^*[x(0)] &= \int_0^t \dot{V}^*(x(\tau)) dt \\
 &\leq \int_0^t (-x^T x) dt \\
 &\leq -\int_0^t c^2 dt \\
 &= -c^2 t
 \end{aligned}$$

Therefore, when $t \rightarrow \infty$, $V^*[x(t)] \leq V^*[x(0)] - c^2 t \rightarrow -\infty$. This contradicts that $V^*[x(t)]$ is positive definite. Consequently, the system (1) is globally asymptotically stable. \square

Remark 1. For matched nonlinear systems, the robust controller can be obtained by solving the optimal cost function $V^*[x(t)]$ from HJB Equation (10). In Section 4, we will use the PI algorithm to solve the HJB equation, which is a difficult partial differential equation.

4. Robust Control of Nonlinear Systems with Mismatched Uncertainties

In this section, we consider the robust control problem when the system (1) does not satisfy the matched condition (2). At this time, the system is a mismatched nonlinear uncertain system. By constructing the appropriate auxiliary system and performance index, the robust control for the mismatched uncertain system is transformed into solving optimal control law of an auxiliary system.

Firstly, the following assumptions are given.

Assumption 3. Suppose that the uncertainty of system (1) satisfies $\Delta f(x) = c(x)h(x)$, $\Delta g(x) = g(x)m(x)$, where $c(x)$ is a known function of appropriate dimensions, $h(x)$ and $m(x)$ are uncertain functions, and $m(x) \geq 0$.

The goal of robust control is to find a control function $u(x)$, which makes the closed-loop system

$$\dot{x} = f(x) + c(x)h(x) + [g(x) + g(x)m(x)]u(x)
 \tag{16}$$

globally asymptotically stable for all uncertainties $h(x)$ and $m(x)$.

In order to obtain the robust controller, an optimal control problem is constructed as follow. For the following auxiliary systems

$$\dot{x} = f(x) + g(x)u + [I - g(x)g(x)^+]c(x)v
 \tag{17}$$

find the controller $u = u(x), v = v(x)$, such that the performance index

$$J(x_0, u) = \int_0^\infty [f_{max}^2(x) + g_{max}^2(x) + \beta^2 x^T x + u^T u + v^T v] dt \tag{18}$$

is minimized, where β is the design parameter, $g(x)^+ = [g^T(x)g(x)]^{-1}g^T(x)$ is a pseudo inverse of the matrix function $g(x)$. Moreover, $f_{max}(x)$ and $g_{max}(x)$ are nonnegative functions and satisfy the conditions

$$\|g(x)^+ c(x)h(x)\| \leq f_{max}(x), \|h(x)\| \leq g_{max}(x) \tag{19}$$

According to the performance index (18), the cost function corresponding to the admissible control $(u(x), v(x))$ is

$$V(x) = \int_t^\infty [f_{max}^2(x) + g_{max}^2(x) + \beta^2 x^T x + u^T u + v^T v] dt \tag{20}$$

The following Bellman equation is obtained by taking the time derivation on both sides of (20)

$$f_{max}^2(x) + g_{max}^2(x) + \beta^2 x^T x + \bar{u}^T \bar{u} + \nabla V^T [f(x) + \bar{g}(x)\bar{u}] = 0, \tag{21}$$

where ∇V is the gradient vector of $V(x)$ with respect to x , $\bar{g}(x) = [g(x), (I - g(x)g(x)^+)c(x)]$, $\bar{u} = [u^T, v^T]^T$.

Defining Hamiltonian functions as

$$H(x, u, \nabla V) = f_{max}^2(x) + g_{max}^2(x) + \beta^2 x^T x + \bar{u}^T \bar{u} + \nabla V^T [f(x) + \bar{g}(x)\bar{u}] \tag{22}$$

Assuming that the minimum value exists and is unique in (22), the optimal control law is given by

$$\begin{aligned} \bar{u}^*(x) &= \begin{bmatrix} u^*(x) \\ v^*(x) \end{bmatrix} = -\frac{1}{2}\bar{g}^T(x)\nabla V^* \\ &= -\frac{1}{2} \begin{bmatrix} g^T(x)\nabla V^* \\ c^T(x)[I - g(x)g(x)^+]^T \nabla V^* \end{bmatrix} \end{aligned} \tag{23}$$

By substituting (23) into (21), the HJB equation is given by

$$f_{max}^2(x) + g_{max}^2(x) + \beta^2 x^T x + \nabla V^{*T} f(x) - \frac{1}{4}\nabla V^{*T} \bar{g}(x)\bar{g}^T(x)\nabla V^* = 0 \tag{24}$$

and the initial value $V^*(0) = 0$.

Remark 2. Generally, the pseudo-inverse of $g(x)$, $g(x)^+$ will exist if its columns are linearly independent when Assumptions 1 and 2 are true [31]. In practical control systems, the function, $g(x)$, is usually column full-rank. Therefore, the pseudo-inverse of the function $g(x)$ is generally satisfied. Furthermore, the pseudo-inverse $g(x)^+$ satisfies $g(x)^+g(x) = I$. However, it does not satisfy $g(x)g(x)^+ = I$. In addition, the auxiliary system constructed above is not a nominal system, but a compensation control term $v(x)$ is added to the nominal system.

If we can choose an appropriate parameter β , the optimal cost $V^*(x)$ can be computed from HJB Equation (24). Then, we can get the optimal control law of system (17) with performance index (18). The following theorem shows that optimal control $u^*(x) = -\frac{1}{2}g^T(x)\nabla V^*$ is a robust controller for uncertain systems.

Theorem 2. Assume that the mismatched uncertain system (16) satisfies Assumptions 4.1, 4.2 and (19). Consider the auxiliary system (17) corresponding to the performance index (18). There

exists a solution $V^*(x)$ in HJB Equation (24) for a selected parameter β , and for a constant β' satisfying $|\beta'| < |\beta|$, such that

$$2v^{*T}(x)v^*(x) \leq \beta'^2 x^T x$$

Then, the optimal control policy $u^*(x) = -\frac{1}{2}g^T(x)\nabla V^*$ can globally asymptotically stabilize the nonlinear uncertain system (16). That is to say, the closed-loop uncertain system $\dot{x}(t) = f(x) + c(x)h(x) + [g(x) + g(x)m(x)]u^*(x)$ is globally asymptotically stable.

Proof. In order to prove the global asymptotic stability of the closed-loop system, $V^*(x)$ is chosen as the Lyapunov function. Considering the performance index (18), $V^*(x)$ is obviously positive, and $V^*(0) = 0$. Taking the time derivative of the function $V^*(x)$ along the system (16), we have

$$\frac{dV^*}{dt} = \nabla V^{*T}[f(x) + c(x)h(x) + g(x)u^*(x)] + \nabla V^{*T}g(x)m(x)u^*(x) \tag{25}$$

Using $u^*(x) = -\frac{1}{2}g^T(x)\nabla V^*$ yields

$$\begin{aligned} \frac{dV^*}{dt} &= \nabla V^{*T}f(x) + \nabla V^{*T}c(x)h(x) + \nabla V^{*T}g(x)u^*(x) - \frac{1}{2}\nabla V^{*T}g(x)m(x)g^T(x)\nabla V^* \\ &\leq \nabla V^{*T}f(x) + \nabla V^{*T}c(x)h(x) + \nabla V^{*T}g(x)u^*(x) \\ &= \nabla V^{*T}[f(x) + g(x)u^*(x) + c(x)h(x)] - \nabla V^{*T}(I - g(x)g(x)^+)c(x)v^*(x) \\ &\quad + \nabla V^{*T}(I - g(x)g(x)^+)c(x)v^*(x) \\ &= \nabla V^{*T}[f(x) + g(x)u^*(x) + (I - g(x)g(x)^+)c(x)v^*(x)] + \nabla V^{*T}g(x)g(x)^+c(x)h(x) \\ &\quad - \nabla V^{*T}(I - g(x)g(x)^+)c(x)v^*(x) + \nabla V^{*T}(I - g(x)g(x)^+)c(x)h(x) \end{aligned}$$

by $u^*(x) = -\frac{1}{2}g^T(x)\nabla V^*$ and $v^*(x) = -\frac{1}{2}c^T(x)(I - g(x)g(x)^+)^T\nabla V^*$,

$$\begin{aligned} \frac{dV^*}{dt} &\leq \nabla V^{*T}f(x) - 2u^{*T}(x)u^*(x) - 2v^{*T}(x)v^*(x) + \nabla V^{*T}g(x)g(x)^+c(x)h(x) \\ &\quad - \nabla V^{*T}(I - g(x)g(x)^+)c(x)v^*(x) + \nabla V^{*T}(I - g(x)g(x)^+)c(x)h(x) \end{aligned}$$

It follows from (24) that

$$\nabla V^{*T}f(x) = -f_{max}^2(x) - g_{max}^2(x) - \beta^2 x^T x + u^{*T}(x)u^*(x) + v^{*T}(x)v^*(x)$$

As a result,

$$\begin{aligned} \frac{dV^*}{dt} &\leq -f_{max}^2(x) - g_{max}^2(x) - \beta^2 x^T x - u^{*T}(x)u^*(x) + v^{*T}(x)v^*(x) \\ &\quad - 2u^{*T}(x)g(x)^+c(x)h(x) - 2v^{*T}(x)h(x) \end{aligned} \tag{26}$$

On the other hand

$$-2u^{*T}(x)g(x)^+c(x)h(x) \leq [g(x)^+c(x)h(x)]^T[g(x)^+c(x)h(x)] + u^{*T}(x)u^*(x)$$

Therefore,

$$\begin{aligned} -u^{*T}(x)u^*(x) - 2u^{*T}(x)g(x)^+c(x)h(x) &\leq [g(x)^+c(x)h(x)]^T[g(x)^+c(x)h(x)] \\ &\leq f_{max}^2(x) \end{aligned} \tag{27}$$

It follows from the basic matrix inequality that

$$\begin{aligned}
 -2v^{*T}(x)h(x) &\leq v^{*T}(x)v^*(x) + h^T(x)h(x) \\
 &\leq v^{*T}(x)v^*(x) + g_{max}^2(x)
 \end{aligned}
 \tag{28}$$

So, it can be obtained from (26)–(28) that

$$\begin{aligned}
 \frac{dV^*}{dt} &\leq -\beta^2 x^T x + 2v^{*T}(x)v^*(x) \\
 &= 2v^{*T}(x)v^*(x) - \beta'^2 x^T x - (\beta^2 - \beta'^2)x^T x \\
 &\leq -(\beta^2 - \beta'^2)x^T x
 \end{aligned}$$

Therefore, by Lyapunov stability theory, the optimal control $u^*(x) = -\frac{1}{2}g^T(x)\nabla V^*$ can make the closed-loop uncertain nonlinear system asymptotically stable. Thus, for a constant $c > 0$, there is a neighborhood $N = \{x : \|x\| < c\}$ such that if the state $x(t)$ enters the neighborhood N , then $x \rightarrow 0$ when $t \rightarrow \infty$. However, $x(t)$ cannot stay out of the domain N forever; otherwise, for all $t > 0$, there is $\|x(t)\| \geq c$. This implies that

$$\begin{aligned}
 V^*[x(t)] - V^*[x(0)] &= \int_0^t \dot{V}^*(x(\tau))d\tau \\
 &\leq \int_0^t -(\beta^2 - \beta'^2)x^T x d\tau \\
 &\leq \int_0^t -(\beta^2 - \beta'^2)c^2 d\tau \\
 &= -(\beta^2 - \beta'^2)c^2 t
 \end{aligned}$$

Hence, when $t \rightarrow \infty$, $V^*[x(t)] \leq V^*[x(0)] - (\beta^2 - \beta'^2)c^2 t \rightarrow -\infty$. This contradicts the positivity of $V^*[x(t)]$. Therefore, system (16) is globally asymptotically stable. We complete the proof. \square

5. Neural Networks Approximation in PI Algorithm

In the first two sections, the robust control of uncertain nonlinear systems was transformed into solving the optimal control of an auxiliary system. However, whether the uncertain system is matched or mismatched, the key issue is how to obtain the solution to corresponding HJB equation. As is well known, it is a nonlinear partial differential equation that is hard to solve. Moreover, solving the HJB equation may lead to the curse of dimensionality [21]. In this section, an online PI algorithm is used to solve the HJB equation iteratively, and neural networks are utilized to approximate the optimal cost in PI algorithm.

5.1. PI Algorithms for Robust Control

For the system with matched uncertainty, the optimal control problem (4) with (5) is considered. For any admissible control, the corresponding cost function can be expressed as

$$\begin{aligned}
 V[x(t)] &= \int_t^\infty [f_{max}^2(x) + x^T x + u^T u]dt \\
 &= \int_t^{t+T} [f_{max}^2(x) + x^T x + u^T u]dt + \int_{t+T}^\infty [f_{max}^2(x) + x^T x + u^T u]dt
 \end{aligned}$$

where $T > 0$ is a selected constant. Therefore, it follows that

$$V[x(t)] = \int_t^{t+T} [f_{max}^2(x) + x^T x + u^T u]dt + V[x(t+T)]
 \tag{29}$$

Based on the integral reinforcement relationship (29) and optimal control (9), the PI algorithm of robust control for matched uncertain nonlinear systems is given below.

The convergence of Algorithm 1 is illustrated as follows. The following conclusion gives an equivalent form of the Bellman Equation (30).

Algorithm 1 PI algorithm of robust control for matched uncertain nonlinear systems

- (1) Select supremum $f_{max}(x)$ to satisfy $\|h(x)\| \leq f_{max}(x)$;
- (2) Initialization: for the nominal nonlinear system (4), select an initial stabilization control $u_0(x)$;
- (3) Policy evaluation: for control input $u_i(x)$, calculate cost $V_i(x)$ from the Bellman equation

$$V_i[x(t)] = \int_t^{t+T} [f_{max}^2(x) + x^T x + u_i^T(x)u_i(x)]dt + V_i[x(t + T)] \tag{30}$$

- (4) Policy improvement: compute the control law $u_{i+1}(x)$ using

$$u_{i+1}(x) = -\frac{1}{2}g^T(x)\nabla V_i. \tag{31}$$

By repeatedly iterating between (30) and (31), until the control input is convergent.

Proposition 1. Suppose that $u_i(x)$ is a stabilization controller of nominal system (4). Then the optimal cost $V_i(x)$ solved from (30) is equivalent to solving the following equation

$$f_{max}^2(x) + x^T x + u_i^T(x)u_i(x) + \nabla V_i[f(x) + g(x)u_i(x)] = 0. \tag{32}$$

Proof. Dividing both sides of (30) by T and finding the limit yields

$$\lim_{T \rightarrow 0} \frac{V_i[x(t + T)] - V_i[x(t)]}{T} + \lim_{T \rightarrow 0} \frac{\int_t^{t+T} [f_{max}^2(x) + x^T x + u_i^T(x)u_i(x)]dt}{T} = 0$$

From the definition of function limit and L'Hospital's rule, we can get

$$\frac{dV_i[x(t)]}{dt} + \lim_{T \rightarrow 0} \frac{d}{dT} \int_t^{t+T} [f_{max}^2(x) + x^T x + u_i^T(x)u_i(x)]dt = 0 \tag{33}$$

It follows that

$$f_{max}^2(x) + x^T x + u_i^T(x)u_i(x) + \nabla V_i[f(x) + g(x)u_i(x)] = 0$$

Thus, we can deduce (32) from (30). On the other hand, along the stable system $\dot{x} = f(x) + g(x)u_i(x)$, finding the time derivative of $V_i(x)$ yields

$$\frac{dV_i[x(t)]}{dt} = \nabla V_i[f(x) + g(x)u_i(x)]$$

Integrating both sides from t to $t + T$, yields

$$V_i[x(t + T)] - V_i[x(t)] = \int_t^{t+T} \nabla V_i[f(x) + g(x)u_i(x)]dt$$

Therefore, we can get the following result from (32)

$$V_i[x(t)] = \int_t^{t+T} [f_{max}^2(x) + x^T x + u_i^T(x)u_i(x)]dt + V_i[x(t + T)]$$

This proves that (32) can deduce (30). □

According to [32–34], if the initial stabilization control policy is given $u_0(x)$, then the follow-up control policy calculated by the iterative relations of (30) and (31) is also a stabilizing control policy, and cost sequence $V_i[x(t)]$ calculated by iteration converges to the

optimal cost. By Proposition 1, it is known that (30) and (32) are equivalent, so the iterative relations (30) and (31) in Algorithm 1 converge to the optimal control and optimal cost.

Similarly, we give a PI algorithm of robust control for nonlinear systems with mismatched uncertainties.

The steps of policy evaluation (34) and policy improvement (35) are iteratively calculated until the policy improvement step does not change the current policy. The optimal cost function is calculated as $V^*(x)$, then $u^*(x) = -\frac{1}{2}g^T(x)\nabla V^*(x)$ is the robust control law.

The convergence proof of Algorithm 2 is similar to Algorithm 1, which will not be repeated here.

Algorithm 2 PI algorithm of robust control for nonlinear systems with mismatched uncertainties

- (1) Decompose the uncertainty properly so that $\Delta f(x) = c(x)h(x)$ and $\Delta g(x) = g(x)m(x)$, select constant parameter β, β' such that $|\beta'| < |\beta|$, and then calculate the nonnegative function $f_{max}(x)$ and $g_{max}(x)$ according to (19);
- (2) For auxiliary system (17), select an initial stabilization control policy $u_0(x)$;
- (3) Policy evaluation: Give a control policy $u_i(x)$, the cost $V_i(x)$ is solved from the following Bellman equation

$$V_i[x(t)] = \int_t^{t+T} [f_{max}^2(x) + g_{max}^2(x) + \beta x^T x + \bar{u}_i^T(x)\bar{u}_i(x)]dt + V_i[x(t+T)]; \tag{34}$$

- (4) Policy improvement: Calculate the control policy using the following update law

$$\bar{u}_{i+1}(x) = -\frac{1}{2}\bar{g}^T(x)\nabla V_i; \tag{35}$$

- (5) Check if the condition $2v^{*T}(x)v^*(x) \leq \beta'^2 x^T x$ is satisfied. Return to step (1) and select the larger constants β and β' when it does not hold.
-

Remark 3. In Step (3) of Algorithm 1 or Algorithm 2, solving $V_i[x(t)]$ from (30) or (34) can be transformed into a least squares problem [17]. By reading enough data online along the system trajectory, the cost function $V(x)$ can be calculated by using the least square principle. However, the cost $V_i[x(t)]$ has no specific expressions. In next subsection, along the system trajectory, online reading of sufficient data on the interval $[t, t + T]$, the cost $V_i[x(t)]$ can be approximated by neural network in PI algorithms. Moreover, implementation of the algorithm does not need to know the system dynamics function $f(x)$.

5.2. Neural Network Approximation of Optimal Cost in PI Algorithm

In the implementation of the PI algorithms, we need to use the data of the nominal system and use the least square method to solve the cost function. However, the cost function of nonlinear optimal control problem has no specific form. Therefore, it is necessary to use neural network structure to approximate the cost function, carry out policy iteration, update weights, and then obtain the approximate optimal cost function. In this subsection, neural network is utilized to approximate the optimal cost in the corresponding HJB equation.

Based on the continuous approximation theory of neural network [35], a single neural network is utilized to approximate the optimal cost in HJB equation. For matched uncertain systems, suppose that the solution $V^*(x)$ of HJB Equation (10) is smooth positive definite, and the optimal cost function on compact set Ω is expressed as

$$V^*(x) = W^T\phi(x) + \varepsilon(x) \tag{36}$$

where $W \in \mathbb{R}^L$ is an unknown ideal weight, and $\phi(\cdot) : \mathbb{R}^n \rightarrow \mathbb{R}^L$ is a linear independent basis vector function. It is assumed that $\phi(x)$ is continuous, $\phi(0) = 0$, and $\varepsilon(x)$ is the error

vector of neural network reconstruction. Thus, the gradient of the optimal cost function (36) can be expressed as

$$\nabla V^* = \frac{\partial V^*}{\partial x} = \nabla \phi^T(x)W + \nabla \varepsilon(x) \tag{37}$$

where $\nabla \varepsilon(x) = \frac{\partial \varepsilon}{\partial x}$. On the basis of approximation property of neural network [35,36], when the number of neurons in hidden layer $L \rightarrow \infty$, the approximation error $\varepsilon(x) \rightarrow 0$, $\nabla \varepsilon(x) \rightarrow 0$. Substituting (36) and (37) into (9), the optimal control is rewritten as follows

$$u^*(x) = -\frac{1}{2}g^T(x)[\nabla \phi^T(x) + \nabla \varepsilon(x)] \tag{38}$$

Assume that \hat{W} is an estimated value of the ideal weight W . Since the ideal weight W in (36) is unknown, the cost function of the i -th iteration in Algorithm 1 is expressed as

$$\hat{V}_i(x) = \hat{W}_i^T \phi(x) \tag{39}$$

Using the approximation of neural network in cost function, the Bellman Equation (30) in Algorithm 1 is rewritten as follows

$$\hat{W}_i^T \phi(x(t)) = \Psi + \hat{W}_i^T \phi(x(t+T)) \tag{40}$$

where $\Psi = \int_t^{t+T} [f_{max}^2(x) + x^T x + u_i^T(x)u_i(x)] dt$. Since the above formula uses neural network to approximate the cost function, the residual error caused by neural network approximation is

$$\varepsilon_i(x(t), T) = \Psi + \hat{W}_i^T \phi(x(t+T)) - \hat{W}_i^T \phi(x(t)) \tag{41}$$

In order to obtain the neural network weight parameters of approximation function, the following objective functions can be minimized in the meaning of least square

$$E = \int_{\Omega} \varepsilon_i(x(t), T)^T \varepsilon_i(x(t), T) dx, \tag{42}$$

that is $\int_{\Omega} \frac{d\varepsilon_i(x(t), T)}{d\hat{W}_i} \varepsilon_i(x(t), T) dx = 0$. Using the definition of inner product, it can be rewritten as

$$\left\langle \frac{d\varepsilon_i(x(t), T)}{d\hat{W}_i}, \varepsilon_i(x(t), T) \right\rangle_{\Omega} = 0 \tag{43}$$

It follows from properties of the internal product that

$$\Phi \hat{W}_i + \langle [\phi(x(t+T)) - \phi(x(t))], \Psi \rangle_{\Omega} = 0 \tag{44}$$

where $\Phi = \langle [\phi(x(t+T)) - \phi(x(t))], [\phi(x(t+T)) - \phi(x(t))]^T \rangle$. Therefore,

$$\hat{W}_i = -\Phi^{-1} \langle [\phi(x(t+T)) - \phi(x(t))], \Psi \rangle_{\Omega} \tag{45}$$

So far, the neural network weight parameters of approximation function $V_i(x)$ can be calculated. Thus, the update control policy can be obtained from (35)

$$\hat{u}_{i+1}(x) = -\frac{1}{2}g^T(x)\nabla \phi^T(x)\hat{W}_i. \tag{46}$$

According to [32,33,35,36], using the policy iteration of RL algorithm, the cost sequence $V_i(x)$ converges to the optimal cost $V^*(x)$, and the control sequence $u_i(x)$ converges to the optimal control function $u^*(x)$.

For mismatched uncertain systems, similar neural network approximation can be used.

6. Simulation Examples

Some simulation examples are presented to verify the feasibility of the robust control design method for uncertain nonlinear systems in this section.

Example 1. Consider the following uncertain nonlinear systems

$$\dot{x}(t) = \begin{bmatrix} 0 & 6 \\ -1 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 0 \\ p_1 x_1 \cos(x_2^2) \end{bmatrix} + \begin{bmatrix} 0 \\ 1 + p_2 x_2^2 \end{bmatrix} u \tag{47}$$

where $x = [x_1, x_2]^T$ is the system state, $\Delta f(x) = \begin{bmatrix} 0 \\ p_1 x_1 \cos(x_2^2) \end{bmatrix}$ is the uncertain disturbance function of the system, $\Delta g(x) = \begin{bmatrix} 0 \\ p_2 x_2^2 \end{bmatrix}$ is input uncertainty function, $p_1 \in [-2, 2]$, $p_2 \in [0, 10]$.

Obviously,

$$\Delta f(x) = g(x)h(x), \Delta g(x) = g(x)m(x) \tag{48}$$

where $g(x) = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$, $h(x) = p_1 x_1 \cos(x_2^2)$, $m(x) = p_2 x_2^2$. Moreover, $|h(x)| \leq |2x_1| = f_{max}(x)$. Thus, the original robust control problem is converted into calculating optimal control law. For nominal system

$$\dot{x}(t) = \begin{bmatrix} 0 & 6 \\ -1 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} u, \tag{49}$$

find the control function u , such that the performance index

$$\begin{aligned} J(x, u) &= \int_0^\infty [f_{max}^2(x) + x^T x + u^T u] dt \\ &= \int_0^\infty [5x_1^2 + x_2^2 + u^2] dt \end{aligned} \tag{50}$$

is minimized.

In order to solve the robust control problem by using Algorithm 1, it is assumed that the optimal cost function $V^*(x)$ has a neural network structure: $V^*(x) = W^T \phi(x)$, where $W = [W_1, W_2, W_3]^T$, $\phi(x) = [x_1^2, x_1 x_2, x_2^2]^T$. The initial weight is taken as $W_0 = [-1, 5, 1.5]^T$, and the initial state of system $x_0 = [2, -0.5]^T$. The neural network weights are calculated iteratively by MATLAB. In each iteration, 10 sets of data samples are collected along the nominal system trajectory to perform the batch least squares problem. After five iterations, the weight converges to $[1.9645, 2.8990, 5.4038]$. The robust control law of uncertain system (47) is $u^* = -1.4495x_1 - 5.4038x_2$. The convergence process of neural network weight is shown in Figure 1, while the changing process of control signal is shown in Figure 2. The uncertain parameter p_1 and p_2 in uncertain system (47) take different values, the state trajectories of the closed-loop system are obtained by the robust control law. Figure 3 shows the trajectory of the closed-loop system when $p_1 = -2, p_2 = 1$. Figure 4 shows the trajectory of the closed-loop system when $p_1 = -1, p_2 = 4$. Figure 5 shows the trajectory of the closed-loop system when $p_1 = 0, p_2 = 7$. Figure 6 shows the trajectory of the closed-loop system is $p_1 = 2, p_2 = 10$. From these figures, we can see that the closed-loop system is stable, which shows the effectiveness of the robust control law.

In this example, because of the linear property of the nominal system, MATLAB software can be used to solve LQR problem directly. With this method, the optimal control is calculated as $u^* = -1.4496x_1 - 5.4038x_2$. It is almost the same as the result of neural network approximation, which shows the validity of Algorithm 1.

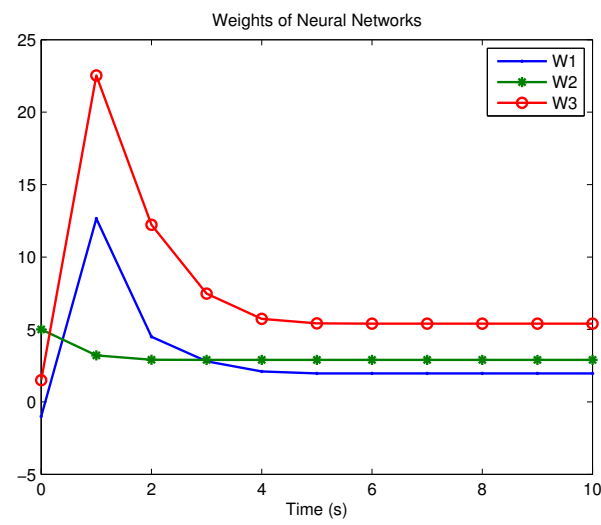


Figure 1. Neural network weight.

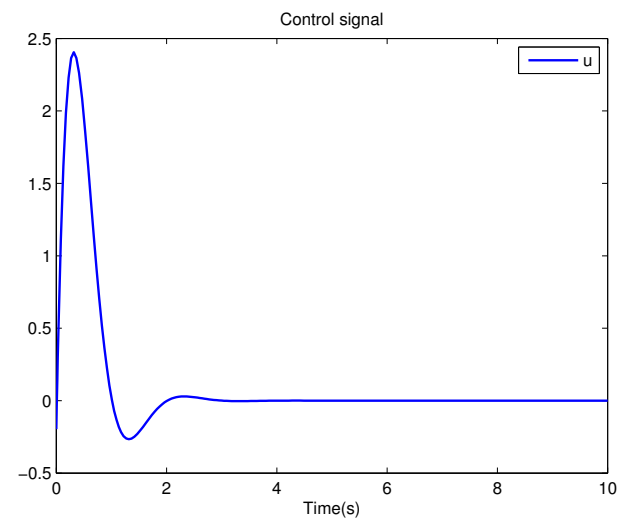


Figure 2. Robust control signal.

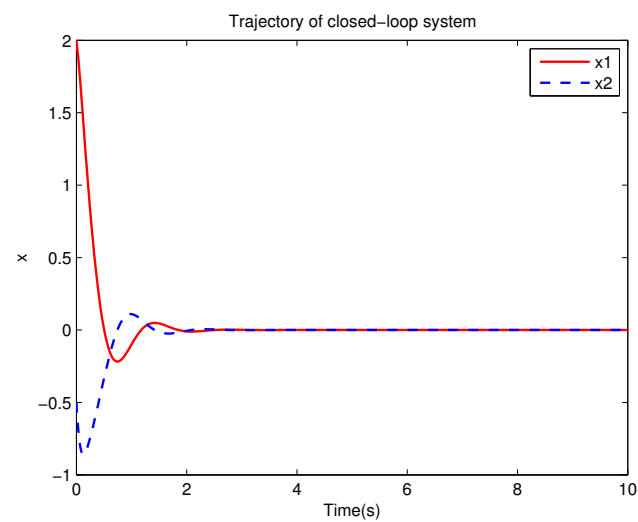


Figure 3. Closed loop system trajectory, $p_1 = -2, p_2 = 1$.

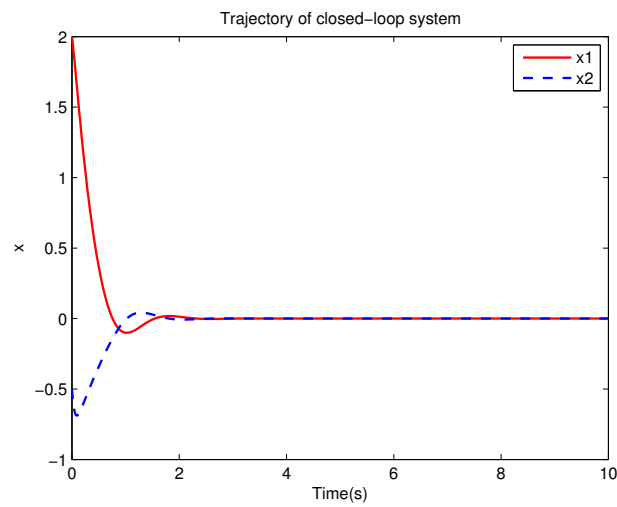


Figure 4. Closed loop system trajectory, $p_1 = -1, p_2 = 4$.

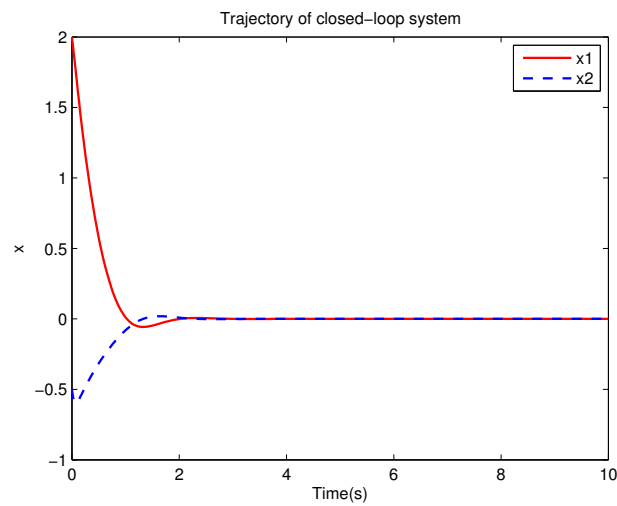


Figure 5. Closed loop system trajectory, $p_1 = 0, p_2 = 7$.

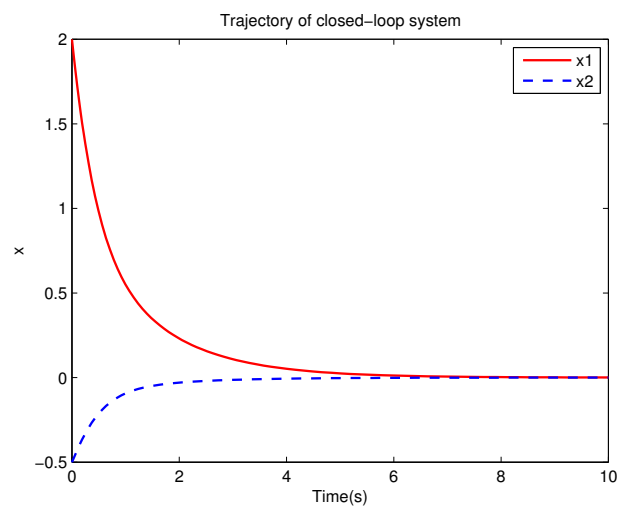


Figure 6. Closed loop system trajectory, $p_1 = 2, p_2 = 10$.

Example 2. Consider the following uncertain nonlinear systems

$$\dot{x}(t) = \begin{bmatrix} 0 & 8 \\ -5 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 0 \\ 0.5 + p_2x_2^2 \end{bmatrix} u + \begin{bmatrix} p_1x_1\cos(x_2^2) + p_3x_2\sin(x_1x_2) \\ 0 \end{bmatrix} \quad (51)$$

where $x = [x_1, x_2]^T$ is the system state, $p_1 \in [-2, 2]$, $p_2 \in [0, 5]$, $p_3 \in [-1, 1]$. Let $\Delta f(x) = \begin{bmatrix} p_1x_1\cos(x_2^2) + p_3x_2\sin(x_1x_2) \\ 0 \end{bmatrix}$, $\Delta g(x) = \begin{bmatrix} 0 \\ p_2x_2^2 \end{bmatrix}$. It is easy to know that the system (51) is a mismatched system. The uncertain disturbance of the system is decomposed as

$$\Delta f(x) = c(x)h(x), \Delta g(x) = g(x)m(x) \quad (52)$$

where, $g(x) = \begin{bmatrix} 0 \\ 0.5 \end{bmatrix}$, $c(x) = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$, $h(x) = p_1x_1\cos(x_2^2) + p_3x_2\sin(x_1x_2)$, $m(x) = p_2x_2^2$. Moreover, $f_{max}(x)$ and $g_{max}(x)$ are calculated as follows.

$$\|g(x)^+c(x)h(x)\| = \|[0 \ 2] \begin{bmatrix} 1 \\ 0 \end{bmatrix} h(x)\| = 0 = f_{max}(x),$$

and

$$\|h(x)\| = \|p_1x_1\cos(x_2^2) + p_3x_2\sin(x_1x_2)\| \leq |2x_1 + x_2| = g_{max}(x)$$

Select the parameter $\beta = 1$. Then the original robust control problem is converted into solving an optimal control problem. For the auxiliary system

$$\dot{x}(t) = \begin{bmatrix} 0 & 8 \\ -5 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 0 & 1 \\ 0.5 & 0 \end{bmatrix} \bar{u}, \quad (53)$$

find the control policy, \bar{u} , such that the following performance index is minimized

$$\begin{aligned} J(x, u) &= \int_0^\infty [f_{max}^2(x) + g_{max}^2(x) + \beta^2x^T x + \bar{u}^T \bar{u}] dt \\ &= \int_0^\infty [5x_1^2 + 2x_2^2 + 4x_1x_2 + \bar{u}^T \bar{u}] dt. \end{aligned} \quad (54)$$

In order to obtain the robust control law by using Algorithm 2, it is assumed that the optimal cost function $V^*(x)$ has a neural network structure: $V^*(x) = W^T \phi(x)$, where $W = [W_1, W_2, W_3]^T$, $\phi(x) = [x_1^2, x_1x_2, x_2^2]^T$. The initial weight is taken as $W_0 = [1, -3, 0.5]^T$, and the initial state of system is chosen as $x_0 = [-2, 0.5]^T$. The neural network weights are calculated iteratively by MATLAB. In each iteration, 10 sets of data samples are collected along the nominal system trajectory to perform the batch least squares problem. After six iterations, the weight converges to $W = [2.8983, -0.6859, 5.2576]^T$. The optimal control of the auxiliary system is calculated as $\bar{u}^* = \begin{bmatrix} 0.1715x_1 - 2.6288x_2 \\ -2.8983x_1 + 0.3429x_2 \end{bmatrix}$. The robust control law of the original uncertain system is $u^* = 0.1715x_1 - 2.6288x_2$. The convergence process of neural network weight is shown in Figure 7, while the changing process of control signal is shown in Figure 8. The uncertain parameters p_1 , p_2 and p_3 in uncertain system (51) take different values, the state trajectories of the closed-loop system are obtained by the robust control law. Figure 9 shows the trajectory of the closed-loop system when $p_1 = -1$, $p_2 = 1$ and $p_3 = 1$. Figure 10 shows the trajectory of the closed-loop system when $p_1 = -1$, $p_2 = 2$ and $p_3 = 0$. Figure 11 shows the trajectory of the closed-loop system when $p_1 = 0.3$, $p_2 = 3$ and $p_3 = -1$. Figure 12 shows the trajectory of the closed-loop system when $p_1 = -2$, $p_2 = 5$ and $p_3 = 1$. From these figures, we can see that the closed-loop system is stable, which shows the effectiveness of the robust control law.

The nominal system is also a linear system, so MATLAB software can be used to solve LQR problem directly. With this method, the optimal control is calculated as $\bar{u}^* = \begin{bmatrix} 0.1713x_1 - 2.6286x_2 \\ -2.8983x_1 + 0.3430x_2 \end{bmatrix}$. It has little difference with the approximate result of neural network, which shows the validity of Algorithm 2.

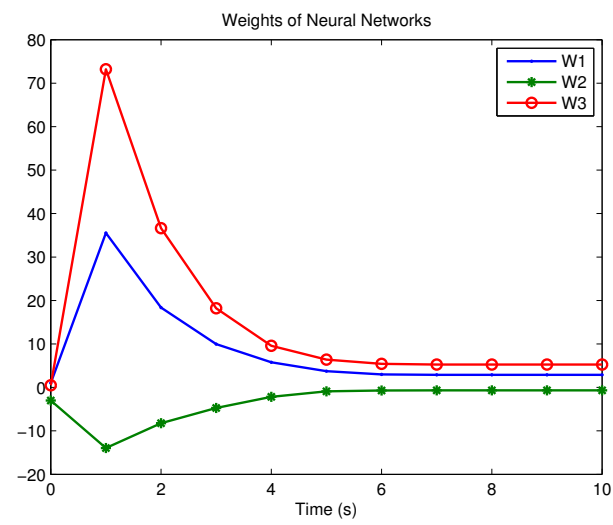


Figure 7. Neural network weight.

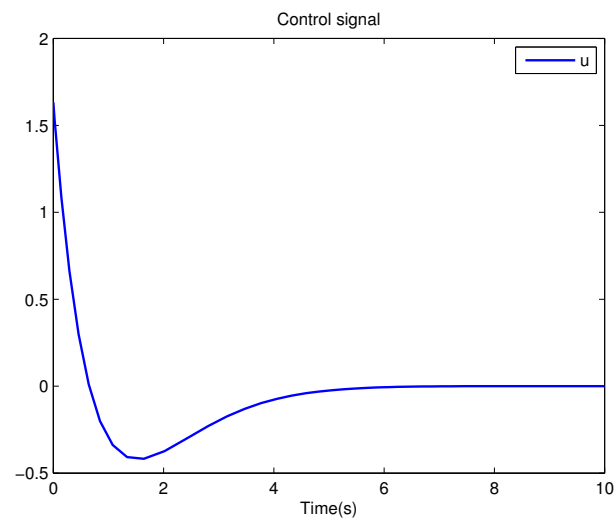


Figure 8. Robust control signal.

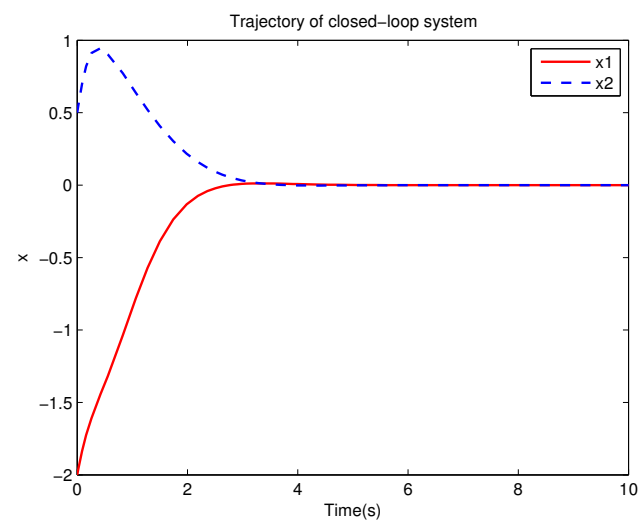


Figure 9. Closed loop system trajectory, $p_1 = -1, p_2 = 2, p_3 = 1$.

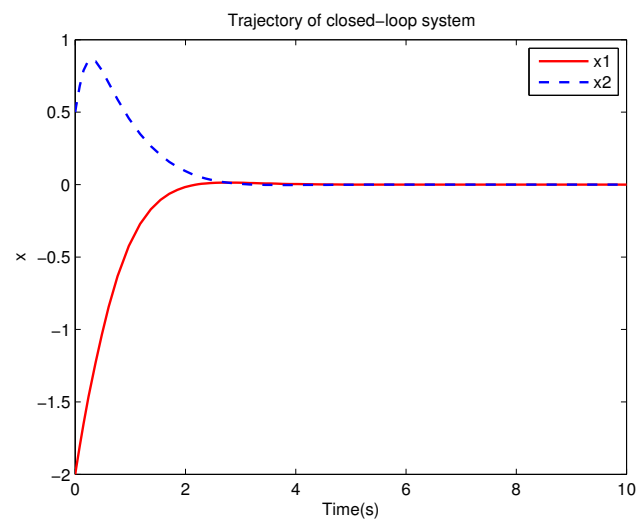


Figure 10. Closed loop system trajectory, $p_1 = -1, p_2 = 2, p_3 = 0$.

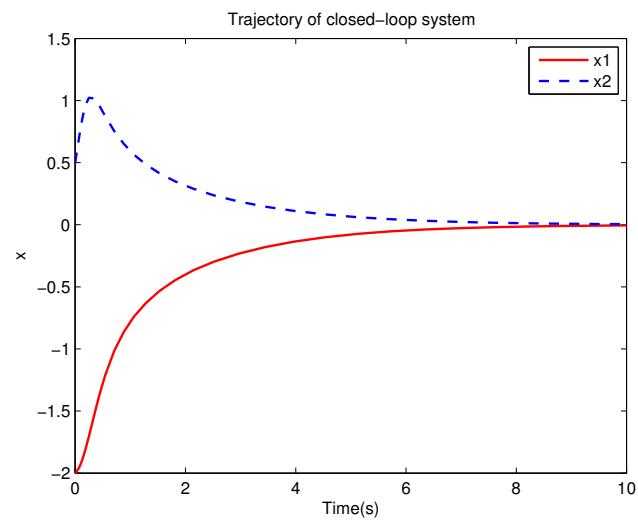


Figure 11. Closed loop system trajectory, $p_1 = 0.3, p_2 = 3, p_3 = -1$.

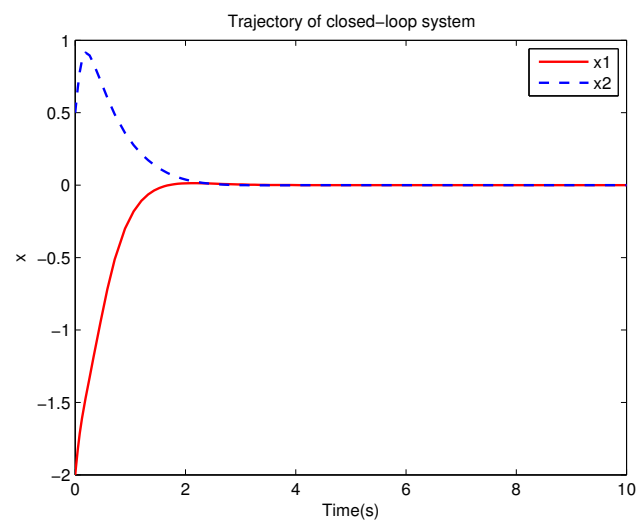


Figure 12. Closed loop system trajectory, $p_1 = -2, p_2 = 5, p_3 = 1$.

The nominal systems corresponding to the above two examples are linear systems. The following is an example with nonlinear nominal system.

Example 3. Consider the following uncertain nonlinear systems

$$\dot{x}(t) = \begin{bmatrix} 1 & 1 \\ -1 & -2 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 0 \\ 2x_1^2 \cos^2(x_2) \end{bmatrix} + \begin{bmatrix} 0 \\ p_1 x_2 \cos^2(x_1) \end{bmatrix} + \begin{bmatrix} 0 \\ 1 + p_2 x_2^2 \end{bmatrix} u \quad (55)$$

where $x = [x_1, x_2]^T$ is the system state, $\Delta f(x) = \begin{bmatrix} 0 \\ p_1 x_2 \cos^2(x_1) \end{bmatrix}$ is the uncertain disturbance function of the system, $\Delta g(x) = \begin{bmatrix} 0 \\ p_2 x_2^2 \end{bmatrix}$ is input uncertainty function, $p_1 \in [-2, 2]$, $p_2 \in [0, 2]$.

Obviously,

$$\Delta f(x) = g(x)h(x), \Delta g(x) = g(x)m(x) \quad (56)$$

where $g(x) = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$, $h(x) = p_1 x_2 \cos^2(x_1)$, $m(x) = p_2 x_2^2$. Moreover, $|h(x)| \leq |2x_2| = f_{\max}(x)$. Thus, the original robust control problem is converted into calculating optimal control law. For nominal system

$$\dot{x}(t) = \begin{bmatrix} 1 & 1 \\ -1 & -2 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 0 \\ 2x_1^2 \cos^2(x_2) \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} u, \quad (57)$$

find the control function u , such that the performance index

$$\begin{aligned} J(x, u) &= \int_0^{\infty} [f_{\max}^2(x) + x^T x + u^T u] dt \\ &= \int_0^{\infty} [x_1^2 + 5x_2^2 + u^2] dt \end{aligned} \quad (58)$$

is minimized.

In order to solve the robust control problem by using Algorithm 1, it is assumed that the optimal cost function $V^*(x)$ has a neural network structure: $V^*(x) = W^T \phi(x)$, where $W = [W_1, W_2, W_3]^T$, $\phi(x) = [x_1^2, x_1 x_2, x_2^2]^T$. The initial weight is taken as $W_0 = [-2, 5, 0.5]^T$, and the initial state of system $x_0 = [2, -0.5]^T$. The neural network weights are calculated iteratively by MATLAB. In each iteration, 10 sets of data samples are collected along the nominal system trajectory to perform the batch least squares problem. After five iterations, the weight converges to $[25.5830, 12.5830, 2.6458]$. The robust control law of uncertain system (55) is $u^* = -6.2915x_1 - 2.6458x_2$. The convergence process of neural network weight is shown in Figure 13, while the changing process of control signal is shown in Figure 14. The uncertain parameter p_1 and p_2 in uncertain system (55) take different values, the state trajectories of the closed-loop system are obtained by the robust control law. Figure 15 shows the trajectory of the closed-loop system when $p_1 = 1$, $p_2 = 0.8$. Figure 16 shows the trajectory of the closed-loop system when $p_1 = -0.5$, $p_2 = 1$. Figure 17 shows the trajectory of the closed-loop system when $p_1 = 1$, $p_2 = 2$. Figure 18 shows the trajectory of the closed-loop system is $p_1 = 2$, $p_2 = 1$. From these figures, we can see that the closed-loop system is stable, which shows the effectiveness of the robust control law.

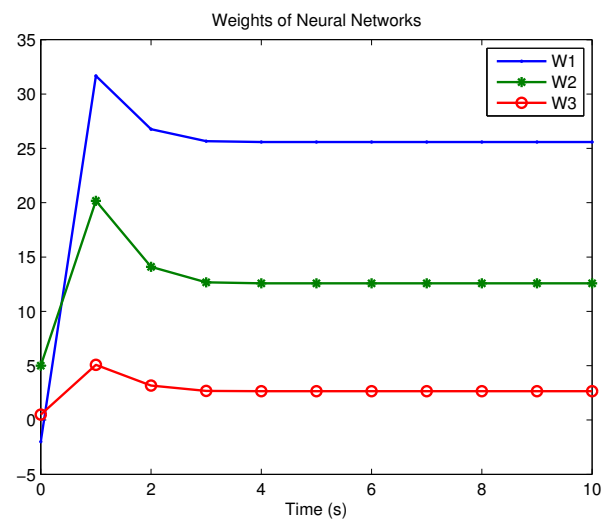


Figure 13. Neural network weight.

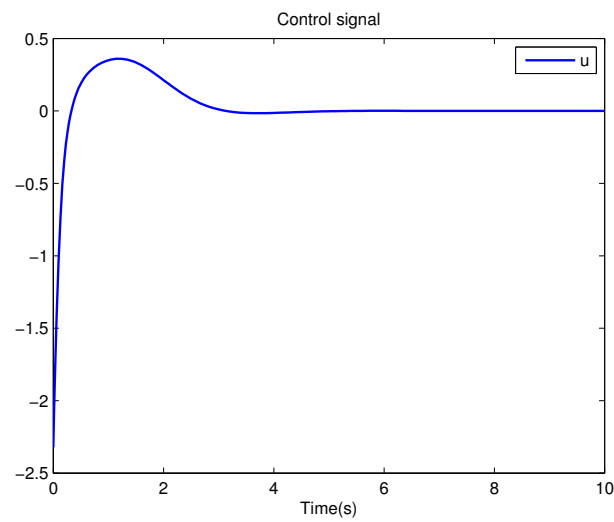


Figure 14. Robust control signal.

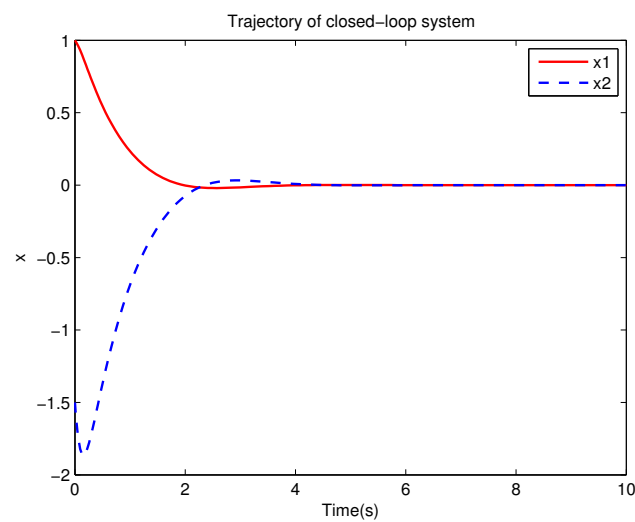


Figure 15. Closed loop system trajectory, $p_1 = 1, p_2 = 0.8$.

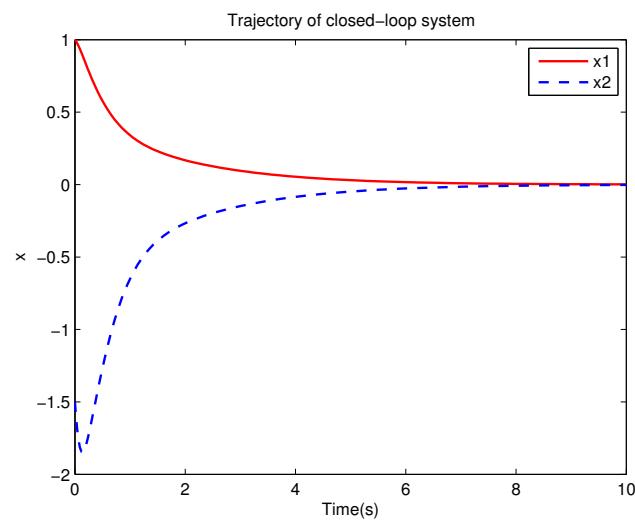


Figure 16. Closed loop system trajectory, $p_1 = -0.5, p_2 = 1$.

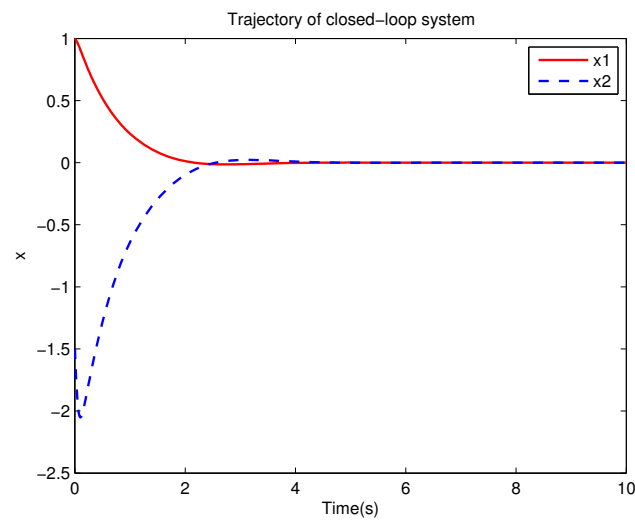


Figure 17. Closed loop system trajectory, $p_1 = 1, p_2 = 2$.

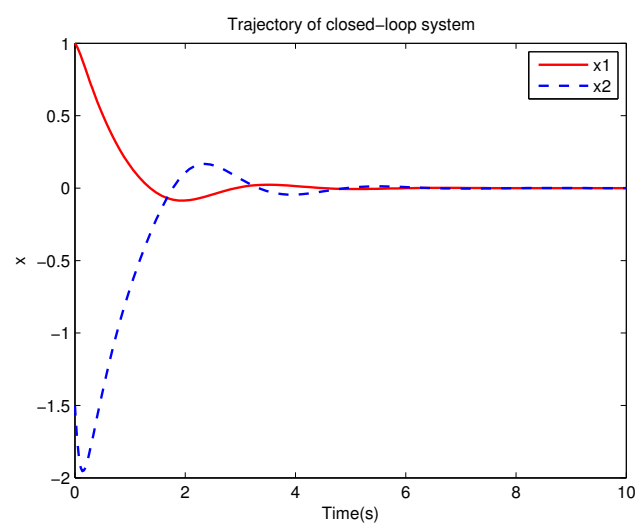


Figure 18. Closed loop system trajectory, $p_1 = 2, p_2 = 1$.

7. Conclusions

In this paper, the PI algorithms in RL are proposed to solve robust control problem for a class of nonlinear continuous time uncertain system. The robust control law is obtained without knowing the internal dynamics of the nominal system. The considered robust control problem is converted into solving an optimal control problem containing a nominal or auxiliary system with a predefined performance index. The online PI algorithms are established to calculate the robust controller of matched and mismatched system. The numerical examples are given to show the availability of the theoretical results. The proposed method may be extended to solve robust tracking problems for some nonlinear systems with uncertainty entering output, which may be the subject of our future research.

Author Contributions: D.X.: investigation, methodology, software; Q.W.: formal analysis; Y.L.: investigation; All the authors contributed equally to the development of the research. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the National Natural Science Foundation of China under Grant No. 61463002.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Data is contained within the article.

Acknowledgments: The authors thank to the Journal editors and the reviewers for their helpful suggestions and comments.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Zhou, K.; Doyle, J.C.; Glover, K. *Robust and Optimal Control*; Prentice Hall: Englewood Cliffs, NJ, USA, 1996.
2. Petersen, I.R.; Hollot, C.V. A riccati equation approach to the stabilization of uncertain linear systems. *Automatica* **1986**, *22*, 397–411. [[CrossRef](#)]
3. Schmitendorf, W.E. Designing stabilizing controllers for uncertain systems using the riccati equation approach. *IEEE Trans. Autom. Control* **1988**, *33*, 376–379. [[CrossRef](#)]
4. Agulhari, C.M.; Oliveira, R.C.; Peres, P.L. Relaxations for Reduced-Order Robust H_∞ Control of Continuous-Time Uncertain Linear Systems. *IEEE Trans. Autom. Control* **2012**, *57*, 1532–1537. [[CrossRef](#)]
5. Jabbari, F.; Schmitendorf, W. A noniterative method for the design of linear robust controllers. *IEEE Trans. Autom. Control* **1990**, *35*, 954–957. [[CrossRef](#)]
6. Tsay, S.C. Robust control for linear uncertain systems via linear quadratic state feedback. *Syst. Control Lett.* **1990**, *15*, 199–205. [[CrossRef](#)]
7. Marino, R.; Tomei, P. Robust stabilization of feedback linearizable time-varying uncertain nonlinear systems. *Automatica* **1993**, *29*, 181–189. [[CrossRef](#)]
8. Shen, T.; Tamura, K. Robust H_∞ control of uncertain nonlinear system via state feedback. *IEEE Trans. Autom. Control* **2002**, *40*, 766–768. [[CrossRef](#)]
9. Teixeira, M.C.M.; Zak, S.H. Stabilizing controller design for uncertain nonlinear systems using fuzzy models. *IEEE Trans. Fuzzy Syst.* **1999**, *7*, 133–142. [[CrossRef](#)]
10. Roy, S.; Kar, I.N.; Lee, J.; Jin, M. Adaptive-Robust Time-Delay Control for a Class of Uncertain Euler–Lagrange Systems. *IEEE Trans. Ind. Electron.* **2017**, *64*, 7109–7119. [[CrossRef](#)]
11. Hosseinzadeh, M.; Yazdanpanah, M.J. Performance enhanced model reference adaptive control through switching non-quadratic Lyapunov functions. *Syst. Control Lett.* **2015**, *76*, 47–55. [[CrossRef](#)]
12. Ma, J.; Xu, S.; Zhuang, G.; Wei, Y.; Zhang, Z. Adaptive neural network tracking control for uncertain nonlinear systems with input delay and saturation. *Int. J. Robust Nonlinear Control* **2020**, *30*, 2593–2610. [[CrossRef](#)]
13. Zhou, L.; She, J.; Zhang, X.M.; Zhang, Z. Additive-state-decomposition based repetitive-control framework for a class of nonlinear systems with multiple mismatched disturbances. *IEEE Trans. Ind. Electron.* **2020**, to be published. [[CrossRef](#)]
14. Liu, L.; Liu, Y.J.; Li, D.; Tong, S.; Wang, Z. Barrier Lyapunov function based adaptive fuzzy FTC for switched systems and its applications to resistance inductance capacitance circuit system. *IEEE Trans. Cybern.* **2020**, *50*, 3491–3502. [[CrossRef](#)]
15. Werbos, P.J. Approximate dynamic programming for real-time control and neural modeling. In *Handbook of Intelligent Control: Neural, Fuzzy, and Adaptive Approaches*; Van Nostrand Reinhold Company: New York, NY, USA, 1992; pp. 493–526.
16. Bhasin, S.; Sharma, N.; Patre, P.; Dixon, W. Asymptotic tracking by a reinforcement learning-based adaptive critic controller. *J. Control Theory Appl.* **2011**, *9*, 400–409. [[CrossRef](#)]

17. Liu, Y.J.; Tang, L.; Tong, S.; Chen, C.P.; Li, D.J. Reinforcement learning design-based adaptive tracking control with less learning parameters for nonlinear discrete-time mimo systems. *IEEE Trans. Neural Netw. Learn. Syst.* **2014**, *26*, 165–176. [[CrossRef](#)] [[PubMed](#)]
18. Vrabie, D.; Pastravanu, O.; Abu-Khalaf, M.; Lewis, F.L. Adaptive optimal control for continuous-time linear systems based on policy iteration. *Automatica* **2009**, *45*, 477–484. [[CrossRef](#)]
19. Modares, H.; Lewis, F.L. Linear quadratic tracking control of partially-unknown continuous-time systems using reinforcement learning. *IEEE Trans. Autom. Control* **2014**, *59*, 3051–3056. [[CrossRef](#)]
20. Kiumarsi, B.; Vamvoudakis, K.G.; Modares, H.; Lewis, F.L. Optimal and autonomous control using reinforcement learning: A survey. *IEEE Trans. Neural Netw. Learn. Syst.* **2018**, *29*, 2042–2062. [[CrossRef](#)] [[PubMed](#)]
21. Lin, F. An optimal control approach to robust control design. *Int. J. Control* **2000**, *73*, 177–186. [[CrossRef](#)]
22. Wang, D.; He, H.; Liu, D. Adaptive critic nonlinear robust control: A survey. *IEEE Trans. Cybern.* **2017**, *47*, 3429–3451. [[CrossRef](#)]
23. Bhasin, S.; Kamalapurkar, R.; Johnson, M.; Vamvoudakis, K.G.; Lewis, F.L.; Dixon, W.E. A novel actor-critic-identifier architecture for approximate optimal control of uncertain nonlinear systems. *Automatica* **2013**, *49*, 82–92. [[CrossRef](#)]
24. Jiang, H.; Zhang, H. Robust control scheme for a class of uncertain nonlinear systems with completely unknown dynamics using data-driven reinforcement learning method. *Neurocomputing* **2018**, *273*, 68–77. [[CrossRef](#)]
25. Zhang, Q.; Zhao, D.; Wang, D. Event-based robust control for uncertain nonlinear systems using adaptive dynamic programming. *IEEE Trans. Neural Netw. Learn. Syst.* **2018**, *29*, 37–50. [[CrossRef](#)] [[PubMed](#)]
26. Yang, X.; He, H.; Zhong, X. Adaptive dynamic programming for robust regulation and its application to power systems. *IEEE Trans. Ind. Electron.* **2018**, *65*, 5722–5732. [[CrossRef](#)]
27. Jia, S.; Jiang, Y.; Li, T.; Du, Y. Learning-Based Optimal Desired Compensation Adaptive Robust Control for a Flexure-Based Micro-Motion Manipulator. *Appl. Sci.* **2017**, *7*, 406. [[CrossRef](#)]
28. Lewis, F.L.; Vrabie, D.; Syrmos, V.L. *Syrmos, Optimal Control*; John Wiley & Sons: New York, NY, USA, 2012.
29. Lin, F. *Robust Control Design: An Optimal Control Approach*; John Wiley & Sons: New York, NY, USA, 2007.
30. Sidorov, N.; Sidorov, D.; Sinitsyn, A.V. *Toward General Theory of Differential-Operator and Kinetic Models*; World Scientific: Singapore, 2020.
31. Ben-Israel, A.; Greville, T.N.E. *Generalized Inverses: Theory and Applications*; Springer Science & Business Media: New York, NY, USA, 2003.
32. Abu-Khalaf, M.; Lewis, F.L. Nearly optimal control laws for nonlinear systems with saturating actuators using a neural network HJB approach. *Automatica* **2005**, *41*, 779–791. [[CrossRef](#)]
33. Vrabie, D.; Lewis, F. Neural network approach to continuous-time direct adaptive optimal control for partially unknown nonlinear systems. *Neural Netw.* **2009**, *22*, 237–246. [[CrossRef](#)] [[PubMed](#)]
34. Xu, D.; Wang, Q.; Li, Y. Optimal Guaranteed Cost Tracking of Uncertain Nonlinear Systems Using Adaptive Dynamic Programming with Concurrent Learning. *Int. J. Control Autom. Syst.* **2020**, *18*, 1116–1127. [[CrossRef](#)]
35. White, H. *Artificial Neural Networks: Approximation and Learning Theory*; Blackwell Publishers, Inc.: Cambridge, UK, 1992.
36. Zhou, Q.; Shi, P.; Tian, Y.; Wang, M. Approximation-Based Adaptive Tracking Control for MIMO Nonlinear Systems with Input Saturation. *IEEE Trans. Cybern.* **2015**, *45*, 2119–2128. [[CrossRef](#)] [[PubMed](#)]