Online policy iterative-based H optimization algorithm for a class of nonlinear systems

DOI:

[10.1016/j.ins.2019.04.027](https://doi.org/10.1016/j.ins.2019.04.027)

Document Version

Accepted author manuscript

[Link to publication record in Manchester Research Explorer](#)

Citation for published version (APA):

He, S., Fang, H., Zhang, M., Liu, F., Luan, X., & Ding, Z. (2019). Online policy iterative-based H optimization algorithm for a class of nonlinear systems. *Information Sciences*. <https://doi.org/10.1016/j.ins.2019.04.027>

Published in:

Information Sciences

Citing this paper

Please note that where the full-text provided on Manchester Research Explorer is the Author Accepted Manuscript or Proof version this may differ from the final Published version. If citing, it is advised that you check and use the publisher's definitive version.

General rights

Copyright and moral rights for the publications made accessible in the Research Explorer are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

Takedown policy

If you believe that this document breaches copyright please refer to the University of Manchester's Takedown Procedures [<http://man.ac.uk/04Y6Bo>] or contact uml.scholarlycommunications@manchester.ac.uk providing relevant details, so we can investigate your claim.

Online policy iterative-based H_∞ optimization algorithm for a class of nonlinear systems

Shuping He^{a,*}, Haiyang Fang^a, Maoguang Zhang^a, Fei Liu^b, Xiaoli Luan^b,
Zhengdao Ding^c

^aKey Laboratory of Intelligent Computing and Signal Processing (Ministry of Education),
School of Electrical Engineering and Automation, Anhui University, Hefei 230601, China

^bKey Laboratory of Advanced Process Control for Light Industry (Ministry of
Education), Institute of Automation, Jiangnan University, Wuxi 214122, China

^cSchool of Electrical and Electronic Engineering, The University of Manchester,
Manchester M13 9PL, UK

Abstract

A novel policy iterative scheme for the design of online H_∞ optimal laws for a class of nonlinear systems is presented. First, neural network-based linear differential inclusion techniques with two multi-layered perceptions are applied to linearize the nonlinear terms. Then, an online partially model-free policy iterative scheme is applied to the linearized system to obtain the design the H_∞ optimal control law. The iterative scheme for the linear H_∞ control problem consists of policy evaluation and policy improvement by means of algebraic Riccati equations. We establish the convergence of the novel policy iterative scheme to the optimal control law. Numerical simulations demonstrating the feasibility and applicability of our design algorithm are provided.

Keywords: nonlinear systems; neural networks; linear differential inclusion; H_∞ optimization; policy iterative schemes.

1. Introduction

Adaptive/approximate dynamic programming (ADP) method has drawn much attention over the past few years [14, 34]. This is mainly due to its char-

*Corresponding author.

Email address: shuping.he@ahu.edu.cn (S. He). (Shuping He)

acterization as a model-free approach, so it can be considered as a free-model approach. In [43], Zhong and He applied the ADP method to continuous-time dynamical systems. Model-free globalized ADP approaches were considered in [44] to solve nonlinear and discrete-time zero-sum game problems. As a kind of ADP iterative techniques, policy iterative (PI) schemes were applied in [13] to study optimal problems related to algebraic Riccati equation (ARE). The design of PI algorithms consists of two parts: policy evaluation and policy improvement. A PI algorithm for solving discrete-time ARE was presented in [21]. In [30], a new online PI algorithm for linear systems was proposed to solve the ARE without the knowledge of the internal dynamics of the system. A similar algorithm for linear dynamics was introduced in [11]. In [5], adaptive optimal control through a novel online PI algorithm was designed for Markov jump systems in the linear case was achieved. An online H_∞ optimal scheme was first considered in [27] and extended by Liu and Wei [16] to learn the infinite optimal solutions for discrete-time nonlinear dynamical systems. The authors in [23] discussed a PI approach to address continuous-time optimal problems. For other possible applications of the PI schemes, the readers are referred to [2, 3, 17, 38, 39].

Compared with linear dynamical systems, the adaptive optimal controller design schemes and the PI algorithms of nonlinear systems are more complicated. In the linear case, designing an optimal controller is equivalent to finding a positive-definite solution of an associated ARE. In general, offline controllers for nonlinear systems can be designed by means of fuzzy or neural network schemes, see for examples [4, 7, 24, 35, 36] and the references therein. Compared with finding ARE solutions for linear dynamics, nonlinear dynamics require finding the associated Hamilton-Jacobi-Bellman (HJB) equation [33] in nonlinear modeling. For example, a PI scheme for adaptive control problems for continuous dynamics was considered in [29] and extended to the discrete case in [16]. In general, the dynamic programming (DP) method [25] also can be used to solve the HJB equation. However, due to the backward-in-time implementation of DP, the conventional methods are all off-time and have a fatal weakness when used for high-order systems. Moreover, computational burden increases dramatically and causes the curse of dimensionality. To address these problems, the reinforcement learning (RL) and the approximate ADP methods have been proposed to study the HJB equations [26, 37, 42].

Considering the fact that the associated ARE of linear systems can be accurately solved by a variety of numerical approaches, we linearize the non-

linear model with an approximate structure in this paper. By linearization, we can avoid solving the nonlinear HJB equations. As a special kind of neural network-based methods, the linear differential inclusion (LDI) technique can be employed to approximate to the nonlinear terms. Tanaka explained the principle of LDI technique elaborately and gave the stability conditions in [28]. Then, Limanond and Si [15] demonstrated an LDI-based control approach for discrete-time systems. Using LDI modeling, the authors in [20] studied the robust estimation of discontinuous neural networks and the receding horizon robust control problem in [1]. The non-autonomous differential inclusion approach is also proposed for distributed optimization based on multi-agent networks in [8]. However, the relevant LDI-based methods to study the optimal problems of nonlinear dynamics are all offline and the knowledge related to the system modeling is also needed. For some optimal control problems in other aspects, the readers can refer to [6, 9, 10, 12, 18, 19, 22, 31, 32, 40, 41].

In our research, a new online PI scheme is considered to address the H_∞ optimal controller design problem of a class of nonlinear systems. Our online PI method does not require the knowledge of the full internal dynamics of the system. Based on the LDI technique, the nonlinear terms are first linearized thus combined with the resulting ARE, the PI scheme is obtained by means of policy evaluation and improvement. We prove the convergence of the PI algorithm. Simulations with nonlinear systems are provided to illustrate the applicability of the new PI scheme.

The main contributions of this paper are highlighted as follows:

- (1) A neural network LDI technique is first used to linearize and approximate the nonlinear terms;
- (2) Combined with an online adaptive policy iteration, we solve the H_∞ optimal control problem for the neural network LDI system;
- (3) A numerical example and a practical one (the inverted pendulum system) are given to show the feasibility of the designed PI algorithms.

We organize the paper in five sections. A description of the H_∞ optimal control problem and the LDI linearization method are given in Section 2. In Section 3, we present a PI scheme for solving the online H_∞ optimal control problem, illustrate the relevant implementation and establish the convergence of our iterative scheme. Section 4 provides two examples to demonstrate the applicability of our method. The conclusion is given in Section 5.

2. Backgrounds and preliminaries

2.1. Problem description

Consider the following class of continuous-time nonlinear systems:

$$\begin{cases} \dot{x}(t) = Ax(t) + Bu(t) + B_1w(t) + f(x(t)) + g(u(t)) \\ y(t) = Cx(t) + Du(t) \\ x(t_0) = x_0, t_0 = 0 \end{cases} \quad (1)$$

where $x(t) \in \mathbb{R}^n$ is the state, $u(t) \in \mathbb{R}^m$ is the control input, $w(t) \in \mathbb{R}^p$ is the unknown disturbance, $y(t) \in \mathbb{R}^z$ is the control output. $f(x(t))$ and $g(u(t))$ are the nonlinear functions which can be sampled and bounded and with $f(0) = 0$ and $g(0) = 0$. A , B , B_1 , C and D are given constant matrices, with compatible dimensions. In addition, we assume that the system (1) is stabilizable.

The solutions of the state-feedback controller design problem via H_∞ performance can be given by $u = -Kx(t)$; for all non-zero bounded disturbance $w(t) \in L_2^P[0, \infty]$, the relation between the control output and the unknown bounded disturbance satisfies a given H_∞ index:

$$J = \|y(t)\|_2 - \gamma\|w(t)\|_2 < 0 \quad (2)$$

where γ is a given disturbance suppression rate and

$$\|y(t)\|_2 = \left[\int_0^\infty y^T(t)y(t)dt \right]^{1/2}, \|w(t)\|_2 = \left[\int_0^\infty w^T(t)w(t)dt \right]^{1/2}. \quad (3)$$

It is well known that the solution of the H_∞ control problem is given by the saddle point stabilizing equilibrium of an associated two-player zero-sum game with cost function given by the infinite horizon domain quadratic function [37]:

$$\begin{aligned} V(x(t), u(t), w(t)) &= \int_0^\infty [\|y(t)\|^2 - \gamma^2\|w(t)\|^2]dt \\ &= \int_0^\infty [x^T(\tau)Qx(\tau) + u^T(\tau)Ru(\tau) - \gamma^2w^T(\tau)w(\tau)]d\tau \end{aligned} \quad (4)$$

where $Q = Q^T \geq 0$, $R = R^T \geq 0$ and $(A, Q^{1/2})$ is supposed to be detectable.

Before proceeding further, we first consider the following linear dynamic model:

$$\begin{cases} \dot{x}(t) = Ax(t) + Bu(t) + B_1w(t) \\ y(t) = Cx(t) + Du(t) \\ x(t_0) = x_0, t_0 = 0 \end{cases} \quad (5)$$

Lemma 1. [37] For the linear system (5), solving the H_∞ control problem (2) is equivalent to finding the Nash equilibrium solutions $(u^*(t), w^*(t))$ given by:

$$\begin{cases} u^*(t) = Kx(t) = R^{-1}B^T P \\ w^*(t) = Lx(t) = \gamma^{-2}B_1^T P \end{cases} \quad (6)$$

where K and L are respectively the gains of the controller and the disturbance and $P = P^T > 0$ is the unique positive solution of the following ARE:

$$A^T P + PA + Q + \gamma^{-2}PB_1B_1^T P - PBR^{-1}B^T P = 0. \quad (7)$$

Lemma 2. [27] Let $K_{(0)}$ and $L_{(0)}$ be the initial stable gains and $P_{(k)}$ be the positive-definite symmetric matrix; then $P_{(k)}$ given in the iterative scheme (8) converges to the unique solution of the ARE (7):

$$(A - BK_{(k)} + B_1L_{(k)})^T P_{(k+1)} + P_{(k+1)}(A - BK_{(k)} + B_1L_{(k)}) = -Q_k \quad (8)$$

where $K_{(k)} = R^{-1}B^T P_{(k-1)}$, $L_{(k)} = \gamma^{-2}B_1^T P_{(k-1)}$, $Q_{(k)} = Q - P_{(k)}(\gamma^{-2}B_1B_1^T + BR^{-1}B^T)P_{(k)}$. $K_{(k)}$ and $L_{(k)}$ are the iteration gains computed recursively. For the solution, $A - BK_{(k)}$ should be Hurwitz and $P_{(k)}$ should converge to P^* in the iteration process, that is, $\lim_{k \rightarrow \infty} P_{(k)} = P^*$, where P^* is the optimal solution of $P_{(k)}$.

Lemma 3. [30] Assume that $K_{(k)}$ is stabilizable. The feedback gain $K_{(k+1)}$ computed using the iterative scheme (8) is also stabilizable when selecting $V(x(t)) = x^T(t)P_{(k)}x(t)$.

Remark 1. Referring to [29] and [37], we shall consider stable pair (A, B) in the initialization of the online solution. For the solution, $A - BK_{(k)}$ is Hurwitz and hence, the pair $(K_{(0)}, L_{(0)})$ is also a stable initial condition. At the same time, the iterative process of Lemma 3 is also proved by the main result in [29].

2.2. Neural network-based LDI linearization

To design an online control policy of the nonlinear system (1), we should first deal with the nonlinear terms. Based on the universal approximation property of neural networks, we provide a class of multi-layered perceptions (MLP's) admitting an LDI state-space representation [15].

Assumption 1. We assume that the nonlinear terms $f(x(t))$ and $g(u(t))$ can be sampled and bounded, i.e.,

$$\begin{cases} \|f(x(t))\|^2 \leq M \\ \|g(u(t))\|^2 \leq M_1 \end{cases} \quad (9)$$

with positive constants M and M_1 .

The nonlinear terms $f(x(t))$ and $g(u(t))$ are modeled by using two L-layered MLP's with the same structure, $N_x(x(t), W_1, W_2, \dots, W_L)$ and $N_u(u(t), V_1, V_2, \dots, V_L)$, respectively. The matrix vectors N_x and N_u can be represented as:

$$\begin{cases} N_x(x(t), W_1, W_2, \dots, W_L) = \Phi_L[W_L \cdots W_2 \Phi_1[W_1 x(t)] \cdots] \\ N_u(u(t), V_1, V_2, \dots, V_L) = \Phi_L[V_L \cdots V_2 \Phi_1[V_1 u(t)] \cdots] \end{cases} \quad (10)$$

where $W_i \in \mathbb{R}^{n_i^x \times n_{i-1}^x}$ and $V_i \in \mathbb{R}^{n_i^u \times n_{i-1}^u}$ are respectively the weight matrices from the $(i-1)$ -th layer to the i -th layer, n_i^s ($i = 1, 2, \dots, L; s \in \{x, u\}$) denotes the number of neurons in the i -th layer, s denotes the MLP's associated with x or u . The input and output of the neural networks are written as $\Phi_i[\zeta] = [\varphi_1(\zeta_1), \varphi_2(\zeta_2), \dots, \varphi_n(\zeta_n)]^T$

We select the neuron basis function $\varphi_i(\zeta_i)$ as:

$$\varphi_i(\zeta_i) = \lambda_i \left(\frac{1 - e^{-\zeta_i/q_i}}{1 + e^{-\zeta_i/q_i}} \right) \quad (11)$$

where $q_i, \lambda_i > 0$ ($i = 1, 2, \dots, n_i$). In (11), we choose $\lambda_i = 1$ and define the derivative vector of the maximum and minimum values of the activation function $\varphi_i(\zeta_i)$ by:

$$S(k, \varphi_i) = \begin{cases} \min_{\zeta_i} \frac{\partial \varphi_i(\zeta_i)}{\partial \zeta_i}, & k = 0 \\ \max_{\zeta_i} \frac{\partial \varphi_i(\zeta_i)}{\partial \zeta_i}, & k = 1 \end{cases} \quad (12)$$

where $\varphi_i(\zeta_i)$ is given by:

$$\varphi_i(\zeta_i) = h_i(0) = h_i(0)S_i(0, \zeta_i) + h_i(1)S_i(1, \zeta_i) \quad (13)$$

where $h_i(k) \geq 0$ ($k = 0, 1$), $\sum_{k=0}^1 h_i(k) = 1$.

According to the back propagation algorithm, one can obtain the optimal approximation weights as:

$$\begin{cases} (W_1^*, W_2^*, \dots, W_L^*) \\ = \arg \min_{W_1, W_2, \dots, W_L} \{ \max_x \| (f(x(t)) - N_x(x(t), W_1, W_2, \dots, W_L)) \| \} \\ (V_1^*, V_2^*, \dots, V_L^*) \\ = \arg \min_{V_1, V_2, \dots, V_L} \{ \max_u \| (g(u(t)) - N_u(u(t), V_1, V_2, \dots, V_L)) \| \} \end{cases} \quad (14)$$

With these optimal weights, we have the following estimates:

$$\begin{cases} \max_x \| (f(x(t)) - N_x(x(t), W^*)) \| \leq \varepsilon_1 \| x(t) \| \\ \max_u \| (g(u(t)) - N_u(u(t), V^*)) \| \leq \varepsilon_2 \| u(t) \| \end{cases} \quad (15)$$

where $\varepsilon_1 > 0, \varepsilon_2 > 0, W^* = (W_1^*, W_2^*, \dots, W_L^*), V^* = (V_1^*, V_2^*, \dots, V_L^*)$.

Here, we suppose that the set of n_i^s dimensional index vectors associated with the i -th L-layered MLP's neural network can be approximated by:

$$\Upsilon_{n_i^s} = \Upsilon_{n_i^s}(\delta_i) = \{ \delta \in \mathcal{R}^{n_i^s} | \delta_i \in \{0, 1\}, i = 1, 2, \dots, n_i^s \} (s \in \{x, u\}). \quad (16)$$

This kind of representation can be viewed as an LDI approximation. To linearize the nonlinear terms $f(x(t))$ and $g(u(t))$, we adopt a compact representation method to describe $N_x(x(t), W_1, W_2, \dots, W_L), N_u(u(t), V_1, V_2, \dots, V_L)$ as:

$$\begin{aligned} N_x(x(t), W_1^*, W_2^*, \dots, W_L^*) &= \Phi_L[W_L^* \dots W_2^* \Phi_1(W_1^* x(t)) \dots] \\ &= \Phi_L \left[W_L^* \dots \Phi_2 \left[W_2^* \left[\begin{array}{c} \sum_{i=0}^1 h_{11}(i) S(i, \zeta_{11})(W_1^* x(t))_1 \\ \vdots \\ \sum_{i=0}^1 h_{1n_1^x}(i) S(i, \zeta_{1n_1^x})(W_1^* x(t))_{n_1^x} \end{array} \right] \dots \right] \right] \\ &= \Phi_L [W_L^* \dots \Phi_2 [W_2^* \sum_{i_{11}=0}^1 \dots \sum_{i_{1n_1^x}=0}^1 h_{11}(i_{11}) \dots h_{1n_1^x}(i_{1n_2^x}) \\ &\quad \cdot \text{diag}[S_{1i}(i_{1i}, \zeta_{1i})] \cdot W_1^* x(t)] \dots] \\ &= \Phi_L [W_L^* \dots \sum_{i_{21}=0}^1 \dots \sum_{i_{2n_2^x}=0}^1 h_{21}(i_{21}) \dots h_{2n_2^x}(i_{2n_2^x}) \cdot \text{diag}[S_{2i}(i_{2i}, \zeta_{2i})] \\ &\quad \cdot [W_2^* \sum_{i_{11}=0}^1 \dots \sum_{i_{1n_1^x}=0}^1 h_{11}(i_{11}) \dots h_{1n_1^x}(i_{1n_1^x}) \cdot \text{diag}[S_{1i}(i_{1i}, \zeta_{1i})] \cdot W_1^* x(t)] \dots] \end{aligned} \quad (17)$$

$$\begin{aligned}
&= \Phi_L [W_L^* \cdots \sum_{i_{21}=0}^1 \cdots \sum_{i_{2n_2^x}=0}^1 \sum_{i_{11}=0}^1 \cdots \sum_{i_{1n_1^x}=0}^1 h_{11}(i_{11}) \cdots h_{1n_1^x}(i_{1n_1^x}) h_{21}(i_{21}) \\
&\quad \cdots h_{2n_2^x}(i_{2n_2^x}) \cdot \text{diag}[S_{2i}(i_{2i}, \zeta_{2i})] \cdot [W_2^* \cdot \text{diag}[S_{1i}(i_{1i}, \zeta_{1i})] \cdot W_1^* x(t)] \cdots] \\
&= \sum_{\sigma_i \in \Theta_i} \zeta_{\sigma_i} A_{\sigma_i}(\sigma_i, \Phi, W^*) x(t),
\end{aligned}$$

$$\begin{aligned}
N_x(u(t), V_1^*, V_2^*, \dots, V_L^*) &= \Phi_L [V_L^* \cdots V_2^* \Phi_1(V_1^* u(t)) \cdots] \\
&= \Phi_L \left[V_L^* \cdots \Phi_2 \left[V_2^* \begin{bmatrix} \sum_{i=0}^1 h_{11}(i) S(i, \zeta_{11})(V_1^* u(t))_1 \\ \vdots \\ \sum_{i=0}^1 h_{1n_1^u}(i) S(i, \zeta_{1n_1^u})(V_1^* u(t))_{n_1^u} \end{bmatrix} \right] \cdots \right] \\
&= \Phi_L [V_L^* \cdots \Phi_2 [V_2^* \sum_{i_{11}=0}^1 \cdots \sum_{i_{1n_1^u}=0}^1 h_{11}(i_{11}) \cdots h_{1n_1^u}(i_{1n_1^u}) \\
&\quad \cdot \text{diag}[S_{1i}(i_{1i}, \zeta_{1i})] \cdot V_1^* u(t)] \cdots] \tag{18} \\
&= \Phi_L [V_L^* \cdots \sum_{i_{21}=0}^1 \cdots \sum_{i_{2n_2^u}=0}^1 h_{21}(i_{21}) \cdots h_{2n_2^u}(i_{2n_2^u}) \cdot \text{diag}[S_{2i}(i_{2i}, \zeta_{2i})] \\
&\quad \cdot [V_2^* \sum_{i_{11}=0}^1 \cdots \sum_{i_{1n_1^u}=0}^1 h_{11}(i_{11}) \cdots h_{1n_1^u}(i_{1n_1^u}) \cdot \text{diag}[S_{1i}(i_{1i}, \zeta_{1i})] \cdot V_1^* u(t)] \cdots] \\
&= \Phi_L [V_L^* \cdots \sum_{i_{21}=0}^1 \cdots \sum_{i_{2n_2^u}=0}^1 \sum_{i_{11}=0}^1 \cdots \sum_{i_{1n_1^u}=0}^1 h_{11}(i_{11}) \cdots h_{1n_1^u}(i_{1n_1^u}) h_{21}(i_{21}) \\
&\quad \cdots h_{2n_2^u}(i_{2n_2^u}) \cdot \text{diag}[S_{2i}(i_{2i}, \zeta_{2i})] \cdot [V_2^* \cdot \text{diag}[S_{1i}(i_{1i}, \zeta_{1i})] \cdot V_1^* u(t)] \cdots] \\
&= \sum_{\eta_i \in \Theta_i} \zeta_{\eta_i} B_{\eta_i}(\eta_i, \Phi, V^*) u(t)
\end{aligned}$$

where L is the number of layers in the network and n is the total number of neurons. Note that $h_{in}(k) \geq 0$ ($k = 0, 1; i = 1, 2, \dots, L$), $h_{in}(0) + h_{in}(1) = 1$ ($n = 1, 2, \dots, n_L^s; s \in \{x, u\}$). $\sum_{\sigma_i \in \Theta_i} \zeta_{\sigma_i}$ and $\sum_{\eta_i \in \Theta_i} \zeta_{\eta_i}$ can be expressed as:

$$\begin{cases} \sum_{\sigma_i \in \Theta_i} \zeta_{\sigma_i} = \sum_{i_{L1}=0}^1 \cdots \sum_{i_{Ln_L^x}=0}^1 \cdots \sum_{i_{11}=0}^1 \cdots \sum_{i_{1n_1^x}=0}^1 h_{11}(i_{11}) \cdots h_{1n_1^x}(i_{1n_1^x}) \\ \quad \cdots h_{L1}(i_{L1}) \cdots h_{Ln_L^x}(i_{Ln_L^x}) = 1 \\ \sum_{\eta_i \in \Theta_i} \zeta_{\eta_i} = \sum_{i_{L1}=0}^1 \cdots \sum_{i_{Ln_L^u}=0}^1 \cdots \sum_{i_{11}=0}^1 \cdots \sum_{i_{1n_1^u}=0}^1 h_{11}(i_{11}) \cdots h_{1n_1^u}(i_{1n_1^u}) \\ \quad \cdots h_{L1}(i_{L1}) \cdots h_{Ln_L^u}(i_{Ln_L^u}) = 1 \end{cases} \tag{19}$$

Then, A_{σ_i} and B_{η_i} can be obtained by the LDI representations (17)-(18):

$$\begin{cases} A_{\sigma_i} = \text{diag}[S_{Li}(\sigma_{Li}, \psi_{Li})] \cdot W_L^* \dots W_2^* \cdot \text{diag}[S_{1i}(\sigma_{1i}, \psi_{1i})] \cdot W_1^* \\ B_{\eta_i} = \text{diag}[S_{Li}(\eta_{Li}, \psi_{Li})] \cdot V_L^* \dots V_2^* \cdot \text{diag}[S_{1i}(\eta_{1i}, \psi_{1i})] \cdot V_1^* \end{cases} \quad (20)$$

According to [1, 15, 20], we can rewrite the nonlinear system (1) as:

$$\begin{cases} \dot{x}(t) = \bar{A}x(t) + \bar{B}u(t) + B_1w(t) \\ y(t) = Cx(t) + Du(t) \\ x(t_0) = x_0, t_0 = 0 \end{cases} \quad (21)$$

where $\bar{A} = A + \sum_{\sigma_i \in \Theta_i} \zeta_{\sigma_i} A_{\sigma_i}$, $\bar{B} = B + \sum_{\eta_i \in \Theta_i} \zeta_{\eta_i} B_{\eta_i}$.

So far, we have used the LDI approximation to model the nonlinear terms. The relevance of such approximation to the H_∞ optimal control problem for the nonlinear system (1) will be studied in Section 3.

3. Main results

In this section, we first propose a PI scheme to solve the linear H_∞ optimal problem. Then, we combine the PI scheme with the LDI technique to address the optimal control problem for a class of nonlinear systems. The PI algorithms proposed in this paper are online and do not require the full knowledge of the system dynamics, i.e., the knowledge of the system matrix A is not needed, while one of the system matrix B is still needed. Therefore, the following algorithms are partially model-free. At the end of the section, we provide a proof of the convergence of the new PI algorithm.

3.1. PI Algorithm for Linear System (5)

Before dealing with the online H_∞ control problem for the nonlinear system (1), we first present the online PI scheme for the linear system (5). The solution of the nonlinear equation (7) can be obtained by means of the linear iterative scheme (8) using an offline algorithm similar to the one introduced in Wu and Luo [37]. Thus, a PI algorithm for solving the linear H_∞ optimal control problem is given in Algorithm 1.

Remark 2. It is obvious that the system dynamic matrix A is not needed in Algorithm 1. The algorithm is thus partially model-free.

Algorithm 1 PI algorithm for linear system (5)

Step 1: Give initial conditions $\{K_{(0)}, L_{(0)}, P_{(0)}\}$

Step 2: Solve the PI equations based on the following two iterative procedures:

1. Online Solving (Policy evaluation):

$$x_t^T P_{(k+1)} x + \Xi(t, T) = \int_t^{t+T} x_\tau^T [Q + K_{(k)}^T R K_{(k)}] x_\tau d\tau + x_{t+T}^T P_{(k+1)} x_{t+T}. \quad (22)$$

2. Gains update (Policy improvement):

$$\begin{cases} K_{(k+1)} = R^{-1} B^T P_{(k)} \\ L(k+1) = \gamma^{-2} B_1^T P_{(k)} \end{cases} \quad (23)$$

where $\Xi(t, T) = \int_t^{t+T} x_\tau^T \gamma^2 L_{(k)}^T L_{(k)} x_\tau d\tau$, $x(t)$ is denoted with x_t . If the solution of $P_{(k)}$ is positive-definite symmetric and satisfies $\|P_{(k+1)} - P_{(k)}\| \leq \alpha$ (α is a small positive real number), stop, output $P_{(k)}$ and go to Step 3; otherwise, go back to Step 2 and continue.

Step 3: Output the optimal gains K^* and P^* .

3.2. A New PI Algorithm

In the following, we present an online PI method for the H_∞ optimal control problem associated with the nonlinear system (1). In designing our new PI method, we shall maintain the partially model-free characteristic of the iterative scheme. The new algorithm combines the LDI technique with the adaptive dynamic programming approach presented in Algorithm 1. The algorithm consists of two parts: linearization and a two-step iterations, i.e., policy evaluation and improvement. Applying Lemma 2 and Lemma 3 to the system (21) leads to the following iterative ARE scheme, which converges to the unique solution to the H_∞ optimal problem associated with (21):

$$(\bar{A} + T\hat{P}_{(k)})^T \hat{P}_{(k+1)} + \hat{P}_{(k+1)}(\bar{A} + T\hat{P}_{(k)}) = -Q + \hat{P}_{(k)} T \hat{P}_{(k)} \quad (24)$$

where $T = \gamma^{-2} B_1 B_1^T - \bar{B} R^{-1} \bar{B}^T$. Similarly, $\bar{A} - \bar{B} K_{(k)}$ should be Hurwitz and $P_{(k)}$ should converge to the optimal value P^* in the iteration process. In the following, we describe the PI algorithm based on the neural network LDI representation in Algorithm 2.

Algorithm 2 PI algorithm based on the neural network LDI

Step 1: Give initial conditions $\{K_{(0)}, L_{(0)}, \hat{P}_{(0)}\}$

Step 2: Using the LDI representations (17) and (18) in each iteration steps, the nonlinear terms in (1) are expressed as:

$$\begin{cases} f(x(t)) = \sum_{\sigma_i \in \Theta_i} \zeta_{\sigma_i} A_{\sigma_i} x(t) \\ g(u(t)) = \sum_{\eta_i \in \Theta_i} \zeta_{\eta_i} B_{\eta_i} u(t) \end{cases} \quad (25)$$

Then, the original system (1) is transformed into (21).

Step 3: Solve the following PI equations based on the two iterative procedures:

1. Online Solving (Policy evaluation):

$$x_t^T \hat{P}_{(k+1)} x + \Xi(t, T) = \int_t^{t+T} x_\tau^T [Q + K_{(k)}^T R K_{(k)}] x_\tau d\tau + x_{t+T}^T \hat{P}_{(k+1)} x_{t+T}. \quad (26)$$

2. Gains update (Policy improvement):

$$\begin{cases} K_{(k+1)} = R^{-1} \bar{B}^T \hat{P}_{(k)} \\ L_{(k+1)} = \gamma^{-2} B_1^T \hat{P}_{(k)} \end{cases} \quad (27)$$

where $\Xi(t, T) = \int_t^{t+T} x_\tau^T \gamma^2 L_{(k)}^T L_{(k)} x_\tau d\tau$, $x(t)$ is denoted with x_t . If the solution of $\hat{P}_{(k)}$ is positive-definite symmetric and satisfies $\|\hat{P}_{(k+1)} - \hat{P}_{(k)}\| \leq \alpha$ (α is a small positive real number), stop, output $\hat{P}_{(k)}$ and go to Step 3; otherwise, go back to Step 3 and continue.

Step 4: Output the optimal gains K^* and L^* .

Remark 3. It is well known that specific assumptions on the initial stability of the system, for example, $(A, B, Q^{1/2})$ is stable-detectable and guarantees the convergence of PI Algorithm 1 to the solution of the linear H_∞ optimal control problem. For the nonlinear system, we linearize the nonlinear terms using the LDI representation and ensure initial stability of the linearized system we have imposed in Assumption 1. This guarantees the boundedness of A_{σ_i}, B_{η_i} at each iteration and leads to the following.

Theorem 4. Assume that $\hat{A} = \bar{A} - \bar{B}K$ is stable. The value $\hat{P}_{(k)}$ obtained from (26) coincides with the one obtained from (24) equals to the solution in iteration procedure (24).

Proof. Inserting $w(t) = Lx(t)$ into (21) yields $\dot{x}(t) = \hat{A}x(t) + B_1w(t)$. Assume that $\hat{A}$ is stabilizable, $Q + \hat{P}_{(k)}T\hat{P}_{(k)} > 0$ and select the Lyapunov function $V(x_t) = x_t^T \hat{P}_{(k)} x_t$. We have:

$$\begin{aligned} \dot{V}(x_t) &= \frac{d}{dt} x_t^T \hat{P}_{(k)} x_t = x_t^T [(A_i + \gamma^{-2} B_1 B_1^T \hat{P}_{(k-1)}) \hat{P}_{(k)} + \\ &\quad \hat{P}_{(k)} (A_i + \gamma^{-2} B_1 B_1^T \hat{P}_{(k-1)})] x_t \\ &= -x_t^T [Q + K_{(k)}^T R K_{(k)} - \gamma^2 L_{(k)}^T L_{(k)}] x_t. \end{aligned} \quad (28)$$

Then, for $\forall T > 0$, it yields:

$$\begin{aligned} x_{t+T}^T \hat{P}_{(k)} x_{t+T} - x_t^T \hat{P}_{(k)} x_t &= \int_t^{t+T} \frac{d}{d\tau} x_\tau^T \hat{P}_{(k)} x_\tau d\tau \\ &= - \int_t^{t+T} x_\tau^T [Q + K_{(k)}^T R K_{(k)} - \gamma^2 L_{(k)}^T L_{(k)}] x_\tau d\tau. \end{aligned} \quad (29)$$

Combing (26) and (27), we can get:

$$(\bar{A} + TP_{(k)})^T P_{(k+1)} + P_{(k+1)} (\bar{A} + TP_{(k)}) = -Q + P_{(k)} T P_{(k)} \quad (30)$$

where $T = \gamma^{-2} B_1 B_1^T - \bar{B} R^{-1} \bar{B}^T$.

Therefore, the solution of the iteration equation (24) is equivalent to solving $\hat{P}$ in the PI equation (26). From the PI algorithm, it does not require the knowledge of the full internal dynamics of the system. It is embedded in $x(t)$ and $x(t+T)$ which can be observed online. This completes the proof. $\square$

Theorem 5. Assume that the triple $(\bar{A}, \bar{B}, Q^{1/2})$ is stabilizable-detectable. The solution of $\hat{P}_{(k)}$ in Algorithm 2 converges to the solutions of the following ARE:

$$(\bar{A} + T\hat{P}_{(k)})^T \hat{P}_{(k+1)} + \hat{P}_{(k+1)}(\bar{A} + T\hat{P}_{(k)}) = -Q + \hat{P}_{(k)}T\hat{P}_{(k)} \quad (31)$$

where $T = \gamma^{-2}B_1B_1^T - \bar{B}R^{-1}\bar{B}^T$.

Proof. Assumption 1 guarantees that the nonlinear terms are bounded. Thus, the linearization terms, i.e., $\sum_{\sigma_i \in \Theta_i} \zeta_{\sigma_i} A_{\sigma_i}$ and $\sum_{\eta_i \in \Theta_i} \zeta_{\eta_i} B_{\eta_i}$ are also bounded. We denote their upper bounds by T_1 and T_2 , respectively. The dynamic matrices based on the LDI representation are bounded and satisfy:

$$\|\bar{A}\| \leq \|A\| + \|T_1\|, \|\bar{B}\| \leq \|B\| + \|T_2\|. \quad (32)$$

Thus, it follows from Theorem 4 that solving equation (26) is equivalent to solving equation (24). Therefore, the positive solution of $\hat{P}_{(k)}$ obtained from Algorithm 2 converges to the solution of the ARE (31), i.e.,

$$\lim_{k \rightarrow \infty} \hat{P}_{(k)} = \hat{P}^* \quad (33)$$

where $\hat{P}^*$ is the solution of the ARE (26). This completes the proof. $\square$

Remark 4. In this paper, we have assumed that the linearization parameters A_{σ_i}, B_{η_i} are bounded by Assumption 1. In the proof of Theorem 5, we used the upper bounds of $\sum_{\sigma_i \in \Theta_i} \zeta_{\sigma_i} A_{\sigma_i}$ and $\sum_{\eta_i \in \Theta_i} \zeta_{\eta_i} B_{\eta_i}$. Obviously, this proves that the solution $\hat{P}_{(k)}$ obtained in Algorithm 2 converges to the solutions of the ARE (31).

3.3. Online implementation

In the online implementation iterative algorithm (26), one can get $K_{(k)}$ from the solution $\hat{P}_{(k)}$. Using the recursive least squares (RLS) method, $V(x_t) = x_t^T \hat{P} x_t$ can be represented as:

$$x_t^T \hat{P}(k) x_t = [\bar{p}(k)]^T \hat{x}_t. \quad (34)$$

Then, we have:

$$\bar{p}_{(k)}^T (\hat{x}_t - \hat{x}_{t+T}) = \int_t^{t+T} x_\tau^T [Q + K_{(k)}^T R K_{(k)} - \gamma^2 L_{(k)}^T L_{(k)}] x_\tau d\tau \quad (35)$$

where $\hat{x}_t = [x_1^2, x_1x_2, \dots, x_1x_n, x_2^2, x_2x_3, \dots, x_2x_n, \dots, x_n^2]$ is the Kronecker product quadratic polynomial basis vector, and

$$\bar{p}_{(k)} = [\hat{P}_{11}, 2\hat{P}_{12}, \dots, 2\hat{P}_{1n}, \hat{P}_{22}, 2\hat{P}_{23}, \dots, \hat{P}_{nn}]^T \quad (36)$$

where $\bar{p}_{(k)}$ is a column valued matrix whose elements depend on the symmetric and positive-definite matrix $\hat{P}_{(k)}$. Following [30], we stack the elements of the diagonal and upper triangular part of $\hat{P}_{(k)}$; the off-diagonal elements are represented as $2\hat{P}_{ij}$ and the diagonal elements are represented as $2\hat{P}_{ii}$ ($i = 1, 2, \dots, n$). Then we know that there are at least N independent elements in the matrix $\hat{P}_{(k)}$, where $N \geq n(n+1)/2$. Therefore, in order to apply the online Algorithm 2 easily, we give the above mathematical expression transformation. When $N \geq n(n+1)/2$, the solution of the policy evaluation algorithm (26) can be derived by the following RLS problem:

$$\bar{p}_{(k)} = (XX^T)^{-1}XY \quad (37)$$

where

$$X = [\hat{X}_1, \hat{X}_2, \dots, \hat{X}_N], \hat{X}_i = \hat{x}_t - \hat{x}_{t+\delta t},$$

$$Y = [D(\hat{X}_1, K_k), D(\hat{X}_2, K_k), \dots, D(\hat{X}_N, K_k)]^T,$$

$$D(\hat{X}_i, K_k) = \int_t^{t+\delta t} x_\tau^T [Q + K_{(k)}^T R K_{(k)} - \gamma^2 L_{(k)}^T L_{(k)}] x_\tau d\tau.$$

We can solve the RLS problem in real-time when we collect enough data points along a single state.

Remark 5. Compared with the results in [16, 17, 33], we do not have to solve the HJB equation by means of LDI linearization, which simplifies the computation and improves the reality of the intractability of the optimal control problems for nonlinear systems. In addition, in contrast to conventional linearization methods (i.e., simply linearize the system around the origin once), our algorithm retains the nonlinear peculiarity of the system and guarantees the computational accuracy to the greatest extent. Based on this points, the combination of neural network LDI technique and the online PI policy provides a new perspective to address such problems. It also shows potential to solve some complicated nonlinear control problems by the designed methods in this paper.

4. Simulation example

Example 1. To further demonstrate the feasibility of our algorithm, we consider a fourth-order nonlinear system with parameters described by:

$$A = \begin{bmatrix} -2.0000 & 1.4000 & -7.8001 & -2.3000 \\ 0.5000 & -6.4123 & -5.2302 & -1.5024 \\ 0.7500 & 11.0000 & -10.2007 & -10.2007 \\ -0.4800 & -5.2032 & 8.2000 & -4.0000 \end{bmatrix}, B = \begin{bmatrix} 0 \\ 0.01 \\ 9.8255 \\ 0 \end{bmatrix}, B_1 = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix},$$

$$Q = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, R = 1, C = [1 \ 0 \ 1 \ 0], D = 0,$$

$$f(x(t)) = \begin{bmatrix} 0 \\ 0 \\ 0 \\ \sin(x_1)\cos(x_1) \end{bmatrix}, g(u(t)) = \begin{bmatrix} 0 \\ 0 \\ 0 \\ e^{-u}\cos(u) \end{bmatrix}.$$

We select two $1 \times 4 \times 1$ neural networks to approximate the nonlinear terms with $q = 0.5, \lambda = 1$. The simulation is conducted by sampling the data at 0.05s. The disturbance attenuation level is selected as $\gamma = 1.5$ and the threshold error α is selected as 10^{-10} .

Executing the online PI algorithm with initial conditions $\hat{P}_{(0)} = 0 * I_4$ and $x_0 = [-0.1 \ -0.2 \ -0.1 \ 0]$, we can obtain the optimal solution:

$$P = \begin{bmatrix} 0.5823 & -0.0404 & 0.0407 & 0.2035 \\ -0.4040 & 0.2560 & 0.0649 & 0.0432 \\ 0.0407 & 0.0649 & 0.1371 & 0.1282 \\ 0.2035 & 0.0432 & 0.1282 & 0.2843 \end{bmatrix}.$$

Then, the H_∞ optimal state-feedback controller gain for the associated nonlinear system (1) is found to be:

$$K = [2.0914 \ 0.9993 \ 2.4130 \ 3.6232].$$

The simulation results are shown in Fig. 1-4. It is seen from the figures that the online PI algorithm with the chosen neural networks is asymptotically stable. For instance, Fig. 1 illustrates the asymptotic stability of the system trajectories $x(t)$. Fig. 2 shows the control input $u(t)$. In Fig. 3,

Figure 1: The system state trajectories $x(t)$ in Example 1.

Figure 2: The control input $u(t)$ in Example 1.

Figure 3: The update process for the elements of the parameters $\hat{P}$ in Example 1.

Figure 4: The system output $y(t)$ in Example 1.

we show the online updating process for the elements of the positive definite optimal solution $\hat{P}$. Each element converges to a fixed optimal value after two iteration. It can be seen from Fig. 4 that the designed optimal H_∞ controller permits the nonlinear system to have better disturbance rejection with a given attenuation level $\gamma = 1.5$.

Example 2. To compare with existing results in the literature, we consider a model of an inverted pendulum [1] given by:

$$\dot{x}(t) = \begin{bmatrix} 0 & 1 \\ 0 & -2 \end{bmatrix} x(t) + \begin{bmatrix} 0 \\ 2 \end{bmatrix} u(t) + \begin{bmatrix} 0 \\ 1 \end{bmatrix} w(t) + \begin{bmatrix} 0 \\ \sin(x_1) \end{bmatrix}.$$

We introduce two $1 \times 3 \times 1$ neural networks to approximate the nonlinear term $\sin(x_1)$ with $q = 0.5$. Executing the online Algorithm 2 with initial conditions as $\hat{P}_{(0)} = 0 * I_2$ and $x_0 = [0.1744 \ 0]^T$, we obtain the optimal solution $\hat{P}$ online at 5-th iterations:

$$P = \begin{bmatrix} 2.3722 & 0.7425 \\ 0.7425 & 0.4551 \end{bmatrix}.$$

Then, the H_∞ optimal state-feedback controller gain is given by:

$$K = [1.4850 \ 0.8902].$$

The simulation results in Figs. 5-7 show that the inverted pendulum system is gradually stable under the action of the H_∞ optimal controller. Thus, the online algorithm is feasible for this practical system. Fig. 5 shows the system state trajectories $x(t)$, which are seen to be asymptotically stable. Fig. 6 gives the control input $u(t)$. In Fig. 7, we show the online updating process for the elements of the positive definite optimal solution $\hat{P}$. Each element converges to a fixed optimal value at 5-th iterations.

The simulation results from these two examples show the effectiveness and feasibility of our design online PI algorithm. Compared with the published results in [25] and [29], our approach gives a direct online algorithm to solve the HJB equation associated with nonlinear systems. But for higher-order systems, solving the HJB equation using our online PI algorithm leads to an increase in the computational burden. In [1], it combines the LMI technique to solve the H_∞ control problem by an offline method. In our design, we gives a combination methods by the LDI technique and the online PI algorithm

Figure 5: The system state trajectories $x(t)$ in Example 2.

Figure 6: The control input $u(t)$ in Example 2.

Figure 7: The update process for the elements of the parameters $\hat{P}$ in Example 2.

method. This PI algorithm can be implemented online to get the relevant optimal solutions.

Remark 6. In the simulation examples, we first apply the LDI of neural networks to linearize the nonlinear terms. Then, combining the online PI algorithm is used to obtain the solutions of equations (26)-(27). The simulation results also show the feasibility and applicability of our design method.

5. Conclusion

A novel online PI scheme to study the H_∞ optimal controller design problem for a class of nonlinear systems without the full knowledge of the system dynamics is presented. The nonlinear terms are first linearized using the LDI technique. Then, the relevant ARE and the H_∞ optimal control law are obtained. We established the convergence proof of the PI algorithm. Two examples are given to illustrate the applicability of the designed online PI algorithm. In the future, we will further study the optimal schemes for more complex nonlinear systems and neural networks.

Acknowledgment. This work was supported in part by the National Natural Science Foundation of China (No. 61673001, 61722306), the State Key Program of National Natural Science Foundation of China (No. 61833007), the Foundation for Distinguished Young Scholars of Anhui Province (No. 1608085J05) and the Key Support Program of University Outstanding Youth Talent of Anhui Province (No. gxydZD2017001).

References

- [1] C. K. Ahn, Receding horizon robust control for nonlinear systems based on linear differential inclusion of neural networks, *Journal of Optimization Theory and Applications* 160 (2) (2014) 659–678.
- [2] X.-H. Chang, J. Xiong, Z.-M. Li, J. H. Park, Quantized static output feedback control for discrete-time systems, *IEEE Transactions on Industrial Informatics* 14 (8) (2018) 3426–3435.
- [3] X.-H. Chang, G.-H. Yang, New results on output feedback H_∞ control for linear discrete-time systems, *IEEE Transactions on Automatic Control* 59 (5) (2014) 1355–1359.
- [4] S. He, Q. Ai, C. Ren, J. Dong, F. Liu, Finite-time resilient controller design of a class of uncertain nonlinear systems with time-delays under asynchronous switching, *IEEE Transactions on Systems, Man, and Cybernetics: Systems* 49 (2) (2019) 281–286.
- [5] S. He, J. Song, Z. Ding, F. Liu, Online adaptive optimal control for continuous-time Markov jump linear systems using a novel policy iteration algorithm, *IET Control Theory and Applications* 9 (10) (2015) 1536–1543.
- [6] S. He, J. Song, F. Liu, Robust finite-time bounded controller design of time-delay conic nonlinear systems using sliding mode control strategy, *IEEE Systems, Man, and Cybernetics: Systems* 48 (11) (2018) 1863–1873.
- [7] W. He, Y. Dong, C. Sun, Adaptive neural impedance control of a robotic manipulator with input saturation, *IEEE Transactions on Systems, Man, and Cybernetics: Systems* 46 (3) (2016) 334–344.

- [8] X. He, T. Huang, J. Yu, C. Li, Y. Zhang, A continuous-time algorithm for distributed optimization based on multiagent networks, *IEEE Transactions on Systems, Man, and Cybernetics: Systems* (2017) DOI: 10.1109/TSMC.2017.2780194.
- [9] B. Jiang, Y. Kao, H. R. Karimi, C. Gao, Stability and stabilization for singular switching semi-Markovian jump systems with generally uncertain transition rates, *IEEE Transactions on Automatic Control* 63 (11) (2018) 3919–3926.
- [10] B. Jiang, H. R. Karimi, Y. Kao, C. Gao, A novel robust fuzzy integral sliding mode control for nonlinear semi-Markovian jump T-S Fuzzy systems, *IEEE Transactions on Fuzzy Systems* 26 (6) (2018) 3594–3604.
- [11] Y. Jiang, Z.-P. Jiang, Computational adaptive optimal control for continuous-time linear systems with completely unknown dynamics, *Automatica* 48 (10) (2012) 2699–2704.
- [12] H. R. Karimi, P. J. Maralani, B. Lohmann, B. Moshiri, H_∞ control of parameter-dependent state-delayed systems using polynomial parameter-dependent quadratic functions, *International Journal of Control* 78 (4) (2005) 254–263.
- [13] D. Kleinman, On an iterative technique for Riccati equation computations, *IEEE Transactions on Automatic Control* 13 (1) (1968) 114–115.
- [14] F. L. Lewis, D. Vrabie, Reinforcement learning and adaptive dynamic programming for feedback control, *IEEE Circuits and Systems Magazine* 9 (3) (2009) 32–50.
- [15] S. Limanond, J. Si, Neural network-based control design: an LMI approach, *IEEE Transactions on Neural Networks* 9 (6) (1998) 1422–1429.
- [16] D. Liu, Q. Wei, Policy iteration adaptive dynamic programming algorithm for discrete-time nonlinear systems, *IEEE Transactions on Neural Networks and Learning Systems* 25 (3) (2014) 621–634.
- [17] D. Liu, Q. Wei, P. Yan, Generalized policy iteration adaptive dynamic programming for discrete-time nonlinear systems, *IEEE Transactions on Systems, Man, and Cybernetics: Systems* 45 (12) (2015) 1577–1591.

- [18] J. Liu, L. Wei, X. Xie, E. Tian, S. Fei, Quantized stabilization for T–S fuzzy systems with hybrid-triggered mechanism and stochastic cyber-attacks, *IEEE Transactions on Fuzzy Systems* 26 (6) (2018) 3820–3834.
- [19] J. Liu, L. Zha, X. Xie, E. Tian, Resilient observer-based control for networked nonlinear T–S fuzzy systems with hybrid-triggered scheme, *Nonlinear Dynamics* 91 (3) (2018) 2049–2061.
- [20] X. Liu, J. Cao, Robust state estimation for neural networks with discontinuous activationseural networks with discontinuous activations, *IEEE Systems, Man, and Cybernetics: Systems* 40 (6) (2010) 1425–1437.
- [21] L.-Z. Lu, W.-W. Lin, An iterative algorithm for the solution of the discrete-time algebraic Riccati equation, *Linear Algebra and Its Applications* 188 (1993) 465–488.
- [22] X. Luan, P. Shi, F. Liu, Given-time multiple frequency control for Markov jump systems based on derandomization, *Information Sciences* (2018) 451–452.
- [23] H. Modares, M.-B. N. Sistani, F. L. Lewis, A policy iteration approach to online optimal control of continuous-time constrained-input systems, *ISA Transactions* 52 (5) (2013) 611–621.
- [24] C. Mu, Z. Ni, C. Sun, H. He, Air-breathing hypersonic vehicle tracking control based on adaptive dynamic programming, *IEEE Transactions on Neural Networks and Learning Systems* 28 (3) (2017) 584–598.
- [25] J. E. Seiffertt, S. Sanyal, D. C. Wunsch, Hamilton-Jacobi-Bellman equations and approximate dynamic programming on time scales, *IEEE Systems Man and Cybernetics: Systems* 38 (4) (2008) 918–923.
- [26] H. Shi, Z. Lin, S. Zhang, X. Li, K. Hwang, An adaptive decision-making method with fuzzy Bayesian reinforcement learning for robot soccer, *Information Sciences* (2018) 436–437.
- [27] J. Song, S. He, Z. Ding, F. Liu, A new iterative algorithm for solving H_∞ control problem of continuous-time Markovian jumping linear systems based on online implementation, *International Journal of Robust and Nonlinear Control* 26 (17) (2016) 3737–3754.

- [28] K. Tanaka, An approach to stability criteria of neural-network control systems, *IEEE Transactions on Neural Networks* 7 (3) (1996) 629–642.
- [29] D. Vrabie, F. L. Lewis, Adaptive optimal control algorithm for continuous-time nonlinear systems based on policy iteration, in: *Proc. 47th IEEE Conference on Decision and Control*, 73–79, 2008.
- [30] D. Vrabie, O. Pastravanu, M. Abu-Khalaf, F. L. Lewis, Adaptive optimal control for continuous-time linear systems based on policy iteration, *Automatica* 45 (2) (2009) 477–484.
- [31] A. K. Waljee, B. Liu, K. Sauder, J. Zhu, S. M. Govani, R. W. Stidham, P. D. R. Higgins, Predicting corticosteroid-free biologic remission with vedolizumab in crohn’s disease, *Inflammatory Bowel Diseases* 24 (6) (2018) 1185–1192.
- [32] A. K. Waljee, B. Liu, K. Sauder, J. Zhu, S. M. Govani, R. W. Stidham, P. D. R. Higgins, Predicting corticosteroid-free endoscopic remission with vedolizumab in ulcerative colitis, *Alimentary Pharmacology and Therapeutics* 47 (6) (2018) 763–772.
- [33] D. Wang, D. Liu, H. Li, Policy iteration algorithm for online design of robust control for a class of continuous-time nonlinear systems, *IEEE Transactions on Automation Science and Engineering* 11 (2) (2014) 627–632.
- [34] F. Wang, H. Zhang, D. Liu, Adaptive dynamic programming: an introduction, *IEEE Computational Intelligence Magazine* 4 (2) (2009) 39–47.
- [35] Y. Wang, H. R. Karimi, H. Shen, Z. Fang, M. Liu, Fuzzy-model-based sliding mode control of nonlinear descriptor systems, *IEEE Transactions on Cybernetics* (2018) DOI: 10.1109/TCYB.2018.2842920.
- [36] Y. Wang, H. Shen, D. Duan, On stabilization of quantized sampled-data neural-network-based control systems, *IEEE Transactions on Systems, Man, and Cybernetics: Systems* 47 (10) (2017) 3124–3135.
- [37] H. Wu, B. Luo, Simultaneous policy update algorithms for learning the solution of linear continuous-time H_∞ state feedback control, *Information Sciences* 222 (2013) 472–485.

- [38] D. Zhai, L. An, X. Li, Q. Zhang, Adaptive fault-tolerant control for nonlinear systems with multiple sensor faults and unknown control directions, *IEEE Transactions on Neural Networks and Learning Systems* 29 (9) (2018) 4436–4446.
- [39] D. Zhai, L. An, D. Ye, Q. Zhang, Adaptive reliable H_∞ static output feedback control against Markovian jumping sensor failures, *IEEE Transactions on Neural Networks and Learning Systems* 29 (3) (2018) 631–644.
- [40] D. Zhang, Z. Xu, H. R. Karimi, Q.-G. Wang, L. Yu, Distributed H_∞ output-feedback control for consensus of heterogeneous linear multiagent systems with aperiodic sampled-data communications, *IEEE Transactions on Industrial Electronics* 65 (5) (2018) 4145–4155.
- [41] D. Zhang, Z. Xu, D. Srinivasan, L. Yu, Leader-follower consensus of multiagent systems with energy constraints: a Markovian system approach, *IEEE Transactions on Systems, Man, and Cybernetics: Systems* 47 (7) (2017) 1727–1736.
- [42] H. Zhang, Q. Wei, D. Liu, An iterative adaptive dynamic programming method for solving a class of nonlinear zero-sum differential games, *Automatica* 47 (1) (2011) 207–214.
- [43] X. Zhong, H. He, An event-triggered ADP control approach for continuous-time system with unknown internal states, *IEEE Transactions on Systems, Man, and Cybernetics: Systems* 47 (3) (2017) 683–694.
- [44] X. Zhong, H. He, D. Wang, Z. Ni, Model-free adaptive control for unknown nonlinear zero-sum differential game, *IEEE Transactions on Cybernetics* 48 (5) (2018) 1633–1646.