IEEE TRANSACTIONS ON

NEURAL NETWORKS AND LEARNING SYSTEMS

A PUBLICATION OF THE IEEE COMPUTATIONAL INTELLIGENCE SOCIETY

http://cis.ieee.org/tnnls

IEEE
Computational
Intelligence
Society

DECEMBER 2012 VOLUME 23 NUMBER 12 ITNNEP (ISSN 2162-237X)

REGULAR PAPERS

BRIEF PAPERS

ANNOUNCEMENTS

IEEE

Neural Network Based Online Simultaneous Policy Update Algorithm for Solving the HJI Equation in Nonlinear H_∞ Control

Huai-Ning Wu and Biao Luo

***Abstract*— It is well known that the nonlinear H_∞ state feedback control problem relies on the solution of the Hamilton–Jacobi–Isaacs (HJI) equation, which is a nonlinear partial differential equation that has proven to be impossible to solve analytically. In this paper, a neural network (NN)-based online simultaneous policy update algorithm (SPUA) is developed to solve the HJI equation, in which knowledge of internal system dynamics is not required. First, we propose an online SPUA which can be viewed as a reinforcement learning technique for two players to learn their optimal actions in an unknown environment. The proposed online SPUA updates control and disturbance policies simultaneously; thus, only one iterative loop is needed. Second, the convergence of the online SPUA is established by proving that it is mathematically equivalent to Newton's method for finding a fixed point in a Banach space. Third, we develop an actor-critic structure for the implementation of the online SPUA, in which only one critic NN is needed for approximating the cost function, and a least-square method is given for estimating the NN weight parameters. Finally, simulation studies are provided to demonstrate the effectiveness of the proposed algorithm.**

***Index Terms*— H_∞ state feedback control, Hamilton–Jacobi–Isaacs equation, neural network, online, simultaneous policy update algorithm.**

I. Introduction

OVER the past few decades, a large number of theoretical results on H_∞ control have been reported [1]–[6]. Although the nonlinear H_∞ control theory has been well developed, the main bottleneck for its practical application is the need to solve the Hamilton–Jacobi–Isaacs (HJI) equation. The HJI equation, similar with the Hamilton–Jacobi–Bellman (HJB) equation of nonlinear optimal control, is a first order nonlinear partial differential equation (PDE), which is difficult or impossible to solve, and may not have global analytic solutions even in simple cases.

In recent years, reinforcement learning (RL) and approximate dynamic programming (ADP) have appeared to be promising techniques for approximately solving nonlinear optimal control problems [7]–[18]. RL [19]–[21] is a kind of machine learning method, which refers to an actor or agent that interacts with its environment and aims to learn the optimal actions, or control policies, by observing their responses from the environment. ADP [20]–[23] solves approximately the dynamic programming problem forward-in-time; thus, it affords a methodology for learning the feedback control actions online in real time based on system performance without necessarily knowing the system dynamics. This overcomes the computational complexity, such as the curse of dimensionality [22] that exists in the classical dynamic programming, which is an offline technique that requires a backward-in-time solution procedure. ADP has many implement structures, such as heuristic dynamic programming (HDP), dual heuristic programming (DHP), globalized DHP, etc., which are widely employed for nonlinear discrete-time systems. In [7], an HDP algorithm was developed to solve the discrete-time HJB equation appearing in infinite horizon discrete-time nonlinear optimal control, and a full proof of convergence was provided. In [12], the near-optimal control problem for a class of nonlinear discrete-time systems with control constraints was solved using a DHP method. Wang *et al.* [14] studied the finite-horizon optimal control problem of discrete-time nonlinear systems and suggested an iterative ADP algorithm to obtain the optimal control law, which makes the performance index function close to the greatest lower bound of all performance indices within a ε-error bound. Policy iteration is one of the most popular RL methods [20] for feedback controller design. In [10] and [11], the optimal control problems of linear and nonlinear continuous-time systems were solved online by policy iteration, respectively. Vamvoudakis and Lewis [13] gave an online synchronous policy iteration algorithm to learn the continuous-time optimal control solution with infinite horizon cost for nonlinear systems with known dynamics, in which an actor and a critic neural network (NN) are involved, and the weights of both NNs tune at the same time instant.

However, it is clear that the HJI equation associated with the nonlinear H_∞ control problem is generally more difficult to solve than the HJB equation appearing in nonlinear optimal control, since the disturbance inputs are additionally reflected in the HJI equation. The main difference between the HJB and HJI equations is that the HJB equation has a "negative semi-definite quadratic term," while the HJI equation has an

Manuscript received January 6, 2012; revised August 29, 2012; accepted August 29, 2012. Date of publication October 12, 2012; date of current version November 20, 2012. This work was supported in part by the National Basic Research Program of China through the 973 Program under Grant 2012CB720003 and in part by the National Natural Science Foundation of China under Grant 61074057, Grant 61121003, and Grant 91016004.

The authors are with the Science and Technology on Aircraft Control Laboratory, School of Automation Science and Electrical Engineering, Beihang University (formerly Beijing University of Aeronautics and Astronautics), Beijing 100191, China (e-mail: whn@buaa.edu.cn; biao.luo@hotmail.com).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TNNLS.2012.2217349

"indefinite quadratic term" [26]. Thus, the methods for the HJB equation may not be directly used to the HJI equation. In [27], the linear H_∞ control problem was considered, in which the H_∞ algebraic Riccati equation (ARE) with an indefinite quadratic term was converted to a sequence of H_2 AREs with a negative semi-definite quadratic term. This paper was subsequently extended to solve the HJI equation for nonlinear systems in [26]. In [28], the solution of the HJI equation was approximated by the Taylor series expansion, and an efficient algorithm was furnished to generate the coefficients of the Taylor series. In [6], it was proven that there exists a sequence of policy iterations on the control input to pursue the smooth solution of the HJI equation, where the HJI equation was successively approximated by a sequence of HJB equations. Then, the methods for solving HJB equations can be used for the solution of the HJI equation. In [29], the HJB equation was successively approximated by a sequence of linear generalized HJB equations, which can be solved by Galerkin's approximation in [30] and [31]. Based on [6] and [29]–[31], policy iteration on the disturbance was used to approximate the HJI equation in [32], where each HJB equation in [6] was further successively approximated by a sequence of generalized HJI equations and solved by Galerkin's approximation. This obviously results in two iterative loops for the solution of HJI equation, i.e., the inner loop solves an HJB equation by iteratively solving a sequence of GHJB equations, and the outer loop solves the HJI equation by iteratively solving a sequence of HJB equations. Following such a thought, the method in [13] was extended to solve the HJI equation in [33] with known dynamics. A policy iteration scheme was also developed in [34] for nonlinear systems with actuator saturation, and its implementation was facilitated on the basis of neurodynamic programming in [35] and [36], where NNs were used for approximating the value function.

Most of the methods mentioned in the above paragraph for solving the HJI equation of H_∞ control problem, such as, [26]–[28], [32]–[36], require full knowledge of the system dynamics. Furthermore, these approaches follow the thought that the HJI equation is first successively approximated with a sequence of HJB equations, and then each HJB equation is solved by the existing methods [26], [32]–[36]. This often brings two iterative loops because the control and disturbance policies are updated at the different iterative steps. Such a procedure may lead to redundant equation solutions (i.e., redundant iterations), and thus waste of sources, resulting in low efficiency. In [37], ADP was used to solve the linear H_∞ control online without the need of internal system dynamics in [37], but it is still a linear special case based on the same procedure as the works in [32]–[36], i.e., it also involves two iterative loops.

In this paper, we propose an online simultaneous policy update algorithm (SPUA) for solving the HJI equation in nonlinear H_∞ state feedback control. The main contributions of this paper include three aspects.

1) Propose an online SPUA, in which the knowledge of internal system dynamics is not required. To the best of our knowledge, this paper may be the first work that uses RL technique for the online H_∞ control design of nonlinear continuous-time systems with unknown internal systems dynamics.
2) The online SPUA updates the control and disturbance policies simultaneously, which needs only one iterative loop rather than two. This is the essential difference between the online SPUA and the existing methods in [32]–[37]. Moreover, the theory of Newton's method in Banach space is introduced to prove the convergence of the online SPUA.
3) Develop an actor-critic structure for nonlinear H_∞ control design without requiring the knowledge of internal system dynamics, where only one critic NN is needed for approximating the cost function and a least-square (LS) method is given to estimate the NN weight parameters.

The rest of this paper is organized as follows. In Section II, we give the problem description. In Section III, we propose the NN-based online SPUA and discuss some related issues. Simulation studies are conducted in Section IV. Finally, a brief conclusion is derived in Section V.

Notations: $\mathbb{R}$, $\mathbb{R}^n$, and $\mathbb{R}^{n\times m}$ are the set of real numbers, the n-dimensional Euclidean space and the set of all real $n \times m$ matrices, respectively. $\|\cdot\|$ denotes the vector norm or matrix norm in $\mathbb{R}^n$ or $\mathbb{R}^{n\times m}$, respectively. For a symmetric matrix M, $M > (\geq) 0$ means that it is a positive (semi-positive) definite matrix. The superscript T is used for the transpose and I denotes the identity matrix of appropriate dimension. $\nabla \triangleq \partial/\partial x$ denotes a gradient operator notation. $L_2[0,\infty)$ is a Banach space, for $\forall w(t) \in L_2[0,\infty)$, $\int_0^\infty \|w(t)\|^2 dt < \infty$. For a column vector function $s(x)$, $\|s(x)\|_\Omega \triangleq \left(\int_\Omega s^T(x)s(x)dx\right)^{1/2}$, $x \in \Omega \subset \mathbb{R}^n$. $H^{m,p}(\Omega)$ is a Sobolev space that consists of functions in space $L_p(\Omega)$ such that their derivatives of order at least m are also in $L_p(\Omega)$.

II. Problem Description

Consider the following partially unknown continuous-time nonlinear system with external disturbance:

$$\dot{x}(t) = f(x) + g(x)u(t) + k(x)w(t) \quad (1)$$

$$z(t) = h(x) \quad (2)$$

where $x \in \Omega \subset \mathbb{R}^n$ is the state, $u \in \mathbb{R}^m$ is the control input and $u(t) \in L_2[0,\infty)$, $w \in \mathbb{R}^q$ is the external disturbance and $w(t) \in L_2[0,\infty)$, and $z \in \mathbb{R}^p$ is the objective output. $f(x)$ is an unknown continuous nonlinear vector function satisfying $f(0) = 0$, which represents the internal system dynamics. $g(x)$, $k(x)$, and $h(x)$ are known continuous vector or matrix functions of appropriate dimensions.

The H_∞ control problem under consideration is to find a state feedback control law $u(t) = u(x(t))$ such that (1) and (2) is closed-loop asymptotically stable, and has L_2-gain less than or equal to γ, that is

$$\int_0^\infty \left(\|z(t)\|^2 + \|u(t)\|_R^2\right) dt \leq \gamma^2 \int_0^\infty \|w(t)\|^2 dt \quad (3)$$

for all $w(t) \in L_2[0,\infty)$, where $\|u(t)\|_R^2 = u^T Ru$, $R > 0$, and $\gamma > 0$ is some prescribed level of disturbance attenuation.

Lemma 1 (see Theorem 16 and Corollary 17 in [6]): Assume (1) and (2) is zero-state observable. Let $\gamma > 0$. Suppose there exists a smooth solution $V^*(x) \geq 0$ to the HJI equation

$$\begin{aligned} G(V^*) = & \left(\nabla V^*(x)\right)^T f(x) + h^T(x)h(x) \\ & -\frac{1}{4}\left(\nabla V^*(x)\right)^T g(x)R^{-1}g^T(x)\nabla V^*(x) \\ & +\frac{1}{4\gamma^2}\left(\nabla V^*(x)\right)^T k(x)k^T(x)\nabla V^*(x) = 0. \end{aligned} \quad (4)$$

Then, the closed-loop system for the state feedback control

$$u(t) = u^*(x(t)) = -\frac{1}{2}R^{-1}g^T(x)\nabla V^*(x) \quad (5)$$

has L_2-gain less than or equal to γ, and the closed-loop system (1), (2), (5) (when $w(t) \equiv 0$) is locally asymptotically stable.

III. NN-Based Online SPUA

An immediate idea for the H_∞ control design of the partially unknown system (1) and (2) is to conduct identification of the system model first, and then model-based approaches can be employed to synthesize the controller. It is noted from Lemma 1 that the nonlinear H_∞ control problem hinges on the solution of the HJI equation (4). However, the HJI equation is a nonlinear PDE that is difficult or impossible to solve, and may not have global analytic solutions even in simple cases. In this section, we propose an online SPUA to solve the HJI equation without requiring the knowledge of the internal system dynamics $f(x)$. Thus, the identification process is avoided.

A. Two-Player Zero-Sum Game

It is well known that the two-player zero-sum differential game theory [1], [26], [27], [33], [35]–[37] has been extensively applied for the H_∞ control problem. Correspondingly, the control input u is a minimizing player and the disturbance w is a maximizing one. Both the H_∞ control problem and the two-player zero-sum differential game rely on the solution of the HJI equation (4). The solution of the H_∞ control problem is the saddle point (u^*, w^*) of the two-player zero-sum game, where u^* and w^* are the optimal control policy and the worst-case disturbance, respectively. Defining the following infinite horizon quadratic cost functional:

$$V(u, w) = \int_0^\infty \left(\|z(t)\|^2 + \|u(t)\|_R^2 - \gamma^2\|w(t)\|^2\right)dt \quad (6)$$

then, the two-player zero-sum game under consideration can be formulated. Given (1) and (2) with two players u and w, and the cost (6), find a saddle point (u^*, w^*) such that

$$V(u^*, w^*) = \min_u \max_w V(u, w) \quad (7)$$

that means

$$V(u^*, w) \leq V(u^*, w^*) \leq V(u, w^*).$$

Define the Hamiltonian of the problem

$$\begin{aligned} H(x, u, w, \nabla V) = & (\nabla V)^T(f + gu + kw) \\ & +h^Th + u^TRu - \gamma^2 w^Tw \end{aligned} \quad (8)$$

then, the HJI equation (4) can also be written as

$$\min_u \max_w H\left(\mathrm{x}, u, w, \nabla V^*\right) = 0 \quad (9)$$

and the saddle point (u^*, w^*) of the game is given as follows:

$$u^*(x) = \arg\min_u H\left(x, u, w^*, \nabla V^*\right) = -\frac{1}{2}R^{-1}g^T\nabla V^* \quad (10)$$

$$w^*(x) = \arg\max_w H\left(x, u^*, w, \nabla V^*\right) = \frac{1}{2}\gamma^{-2}k^T\nabla V^*. \quad (11)$$

B. Online SPUA

It follows from (6) that, given arbitrary control action $u(t)$ and disturbance signal $w(t)$ with initial system state $x(t)$, the cost function is

$$V(x(t)) = \int_t^\infty \left(\|h(x(\tau))\|^2 + \|u(\tau)\|_R^2 - \gamma^2\|w(\tau)\|^2\right)d\tau$$

which can be rewritten as

$$\begin{aligned} V(x(t)) = & \int_t^{t+\Delta t} \left(\|h(x(\tau))\|^2 + \|u(\tau)\|_R^2 - \gamma^2\|w(\tau)\|^2\right)d\tau \\ & +V(x(t+\Delta t)). \end{aligned} \quad (12)$$

Based on (12), we propose the online SPUA (as shown in Algorithm 1) for finding the solution $V^*(x)$ of the HJI equation (4).

Remark 1: The online SPUA follows the basic procedure of policy iteration in RL, which involves policy evaluation (in Step 2) and policy improvement (in Step 3). Hence, it can also be viewed as an RL technique for two players to learn their optimal actions in the unknown environment.

Algorithm 1 Online SPUA

Step 1: Given an initial function $V^{(0)} \in \mathbb{V}_0$ ($\mathbb{V}_0 \subset V$ is determined by Lemma 5), let $u^{(0)} = -\frac{1}{2}R^{-1}g^T\nabla V^{(0)}$, $w^{(0)} = \frac{1}{2}\gamma^{-2}k^T\nabla V^{(0)}$, and $i = 0$.

Step 2: With policies $u^{(i)}$ and $w^{(i)}$, solve the following equation for the cost function $V^{(i+1)}$:

$$\begin{aligned} & V^{(i+1)}(x(t)) \\ & = \int_t^{t+\Delta t}\left(\left\|h(x(\tau))\right\|^2 + \left\|u^{(i)}(\tau)\right\|_R^2 - \gamma^2\left\|w^{(i)}(\tau)\right\|^2\right)d\tau \\ & \quad +V^{(i+1)}(x(t+\Delta t)). \end{aligned} \quad (13)$$

Step 3: Update the control and disturbance policies by

$$\begin{aligned} u^{(i+1)} & = \arg\min_u H\left(x, u, w^{(i)}, \nabla V^{(i+1)}\right) \\ & = -\frac{1}{2}R^{-1}g^T\nabla V^{(i+1)} \end{aligned} \quad (14)$$

$$\begin{aligned} w^{(i+1)} & = \arg\max_w H\left(x, u^{(i)}, w, \nabla V^{(i+1)}\right) \\ & = \frac{1}{2}\gamma^{-2}k^T\nabla V^{(i+1)}. \end{aligned} \quad (15)$$

Step 4: Set $i = i + 1$. If $\left\|V^{(i)} - V^{(i-1)}\right\|_\Omega \leq \varepsilon$ (ε is a small positive real number), stop and output $V^{(i)}$ as the solution of the HJI equation (4) (i.e., $V^* = V^{(i)}$), else, go back to Step 2 and continue.

Remark 2: Although the procedure of policy iteration is also included in the methods of [32]–[36], there are two main differences between these methods and the online SPUA. 1) The methods in [32]–[36] are model-based ones, which require the full knowledge of the system dynamics, while the online SPUA does not require the internal system dynamics $f(x)$. 2) The methods in [32]–[36] update the control and disturbance policies at different iterative steps (i.e., one player updates its policy while the other remains invariant), which brings two iterative loops. In contrast, the online SPUA updates the control and disturbance policies at the same iterative step, in which only one iterative loop is needed. This means that the online SPUA is essentially different from the methods in [32]–[36]. The methods in [32]–[36] are based on the same procedure as in [32], their convergence can be directly guaranteed by the results in [6] and [32]. However, new tools are needed for the online SPUA to establish its convergence.

Remark 3: Notice that the SPUA avoids the identification of $f(x)$ whose information is embedded in the online measurement of the states $x(t)$ and $x(t+\Delta t)$, and evaluation of the cost $\int_t^{t+\Delta t}\left(\|h\left(x(\tau)\right)\|^2 + \|u(\tau)\|_R^2 - \gamma^2\|w(\tau)\|^2\right)d\tau$. That is to say, the lack of knowledge about $f(x)$ does not have any impact on the online SPUA to obtain the equilibrium solution. Thus, the resulting equilibrium behavior policies of the two players will not be affected by any errors between the dynamics of a model of the system and the dynamics of the real system.

C. Convergence of Online SPUA

In this section, we will prove the convergence of the online SPUA. Namely, we want to show that the solution of equation (13) converges to the solution of HJI equation (4) when i goes to infinity. Just as mentioned in Remark 2, the online SPUA is essentially different from the algorithm framework in [32]–[36]. Hence, its convergence proof is also different.

To this end, let us consider such a Banach space $\mathbb{V} \subset \{V(x) | V(x) : \Omega \to \mathbb{R}, V(0) = 0\}$ equipped with a norm $\|\cdot\|_\Omega$, and consider the mapping $G : \mathbb{V} \to \mathbb{V}$ defined in (4). Define a mapping $T : \mathbb{V} \to \mathbb{V}$ as follows:

$$TV = V - \left(G'(V)\right)^{-1} G(V) \tag{16}$$

where $G'(V)$ is the Fréchet derivative of $G(\cdot)$ at point V. It should be noticed that both $G'(V)$ and $\left(G'(V)\right)^{-1}$ are operators on Banach space $\mathbb{V}$.

The Fréchet derivative is often difficult to compute directly, thus we introduce the Gâteaux derivative.

Definition 1 (Gâteaux Derivative) [38]: Let G: $\mathbb{U}(V) \subseteq \mathbb{X} \to \mathbb{Y}$ be a given map, with $\mathbb{X}$ and $\mathbb{Y}$ Banach spaces. Here, $\mathbb{U}(V)$ denotes a neighborhood of V. The map G is Gâteaux differentiable at V if there exists a bounded linear operator $L : \mathbb{X} \to \mathbb{Y}$ such that

$$G(V + sW) - G(V) = sL(W) + o(s), \qquad s \to 0 \tag{17}$$

for all W with $\|W\|_\Omega = 1$ and all real numbers s in some neighborhood of zero, where $\lim_{s\to 0}(o(s)/s) = 0$. L is called the Gâteaux derivative of G at V. The Gâteaux differential at V is defined by $L(W)$.

From (17), the Gâteaux differential at V can be defined equivalently through the following expression [38]:

$$L(W) = \lim_{s\to 0}\frac{G(V + sW) - G(V)}{s}. \tag{18}$$

Equation (18) gives a method to compute Gâteaux derivative, rather than Fréchet derivative required in (16). Thus, we introduce the following lemma to give the relationship between them.

Lemma 2 [38]: If G' exists as Gâteaux derivative in some neighborhood of V, and if G' is continuous at V, then $L = G'(V)$ is also an Fréchet derivative at V.

Now, it follows from Lemma 2 that we can compute the Fréchet derivative $G'(V)$ in (16) via (18). We have the following result.

Lemma 3: Let $G : \mathbb{V} \to \mathbb{V}$ be a mapping defined as (4), then, for $\forall V \in \mathbb{V}$, the Fréchet differential of G at V is

$$\begin{aligned} G'(V)W = L(W) &= (\nabla W)^T f - \frac{1}{4}(\nabla W)^T gR^{-1}g^T\nabla V \\ &\quad -\frac{1}{4}(\nabla V)^T gR^{-1}g^T\nabla W + \frac{1}{4\gamma^2}(\nabla W)^T kk^T\nabla V \\ &\quad +\frac{1}{4\gamma^2}(\nabla V)^T kk^T\nabla W. \end{aligned} \tag{19}$$

Proof: See Appendix.

The following theorem provides an interesting result, in which we discover that the online SPUA is mathematically equivalent to Newton's iteration in a Banach space $\mathbb{V}$.

Theorem 1: Let T be a mapping defined by (16). Then, the iteration from (13) to (15) is equivalent to the following Newton's iteration with (14) and (15):

$$V^{(i+1)} = TV^{(i)}, \qquad i = 0, 1, 2, \ldots \tag{20}$$

Proof: See Appendix.

Under some proper assumptions, Newton's iteration (20) can converge to the unique solution of the fixed-point equation $TV^* = V^*$, that is, the solution of equation $G(V^*) = 0$. The convergence of Newton's method is guaranteed by the following Kantorovtich's theorem [39], [40].

Lemma 4 (Kantorovtich's Theorem): Assume for some $V^{(0)} \in \mathbb{V}_1 \subset \mathbb{V}$ such that $\left(G'(V^{(0)})\right)^{-1}$ exists and that:

1) $$\left\|\left(G'(V^{(0)})\right)^{-1}\right\|_\Omega \le B_0; \tag{21}$$

2) $$\left\|\left(G'(V^{(0)})\right)^{-1} G(V^{(0)})\right\|_\Omega \le \eta; \tag{22}$$

3) for all $V^{(1)}, V^{(2)} \in \mathbb{V}_1$,
$$\left\|G'(V^{(1)}) - G'(V^{(2)})\right\|_\Omega \le K\left\|V^{(1)} - V^{(2)}\right\|_\Omega \tag{23}$$

with $h = B_0 K\eta \le 1/2$. Let

$$\mathbb{V}_2 = \left\{V | \left\|V - V^{(0)}\right\|_\Omega \le \sigma\right\}, \text{where } \sigma = \frac{1 - \sqrt{1 - 2h}}{h}\eta. \tag{24}$$

Now, if $\mathbb{V}_2 \subset \mathbb{V}_1$, then, the sequence $\{V^{(i)}\}$ given in (20) is well defined, remains in $\mathbb{V}_2$, and converges to $V^* \in \mathbb{V}_2$ such that $G(V^*) = 0$. In addition

$$\left\|V^* - V^{(i)}\right\|_\Omega \le \frac{\eta}{h}\frac{\left(1-\sqrt{1-2h}\right)^{2^i}}{2^i}, \qquad i = 0, 1, 2, \dots \tag{25}$$

It is observed from Lemma 4 that the $\mathbb{V}_1$ must be suitably chosen. The following lemma gives a method to determine a $\mathbb{V}_0$ satisfying $\mathbb{V}_0 \subset \mathbb{V}_1$, so that $\mathbb{V}_0$ conversely guarantees the hypotheses of Lemma 4.

Lemma 5 [41]: Suppose $V^* \ge 0$ is the solution of HJI equation $G(V^*) = 0$. If $\|\left(G'(V^*)\right)^{-1}\|_\Omega \le B^*$, and

$$\mathbb{V}_3 = \left\{ V \mid \left\|V - V^*\right\|_\Omega \le \left(\frac{1}{B^*K}\right)\right\} \subset \mathbb{V}_1 \tag{26}$$

then, the hypotheses of Lemma 4 are satisfied. That is, for each $V^{(0)} \in \mathbb{V}_0$, $h \le 1/2$, conditions (21) and (22) hold with

$$B_0 = \frac{B^*}{1 - B^*K\left\|V^{(0)} - V^*\right\|_\Omega} \ge \left\|\left(G'(V^{(0)})\right)^{-1}\right\|_\Omega$$

and

$$\begin{aligned}\eta &= \frac{1 - \frac{1}{2}B^*K\left\|V^{(0)} - V^*\right\|_\Omega}{1 - B^*K\left\|V^{(0)} - V^*\right\|_\Omega}\left\|V^{(0)} - V^*\right\|_\Omega \\ &\ge \left\|\left(G'(V^{(0)})\right)^{-1} G(V^{(0)})\right\|_\Omega\end{aligned}$$

where

$$\mathbb{V}_0 = \left\{ V \mid \left\|V - V^*\right\|_\Omega \le \frac{(2-\sqrt{2})}{(2B^*K)}\right\}. \tag{27}$$

Lemmas 4 and 5 imply that if $V^{(0)}$ is chosen in $\mathbb{V}_0$ defined by (27) ($\mathbb{V}_0$ is a neighborhood of nonnegative definite solution $V^* \ge 0$), the online SPUA, i.e., Newton's method, is bound to converge to the fixed point of (16), i.e., the solution of HJI equation (4), and the error bound is given by (25).

Remark 4: Theorem 1 shows that the sequence $\{V^{(i)}\}$ generated by the online SPUA is equivalent to the Newton sequence obtained by (20), the convergence of which can be guaranteed by Lemma 4. Therefore, the sequence $\{V^{(i)}\}$ also converges to the solution V^* of HJI equation (4), i.e.,$V^{(i)} \to V^*$, when $i \to \infty$. Once V^* is obtained, the saddle point (u^*, w^*) can be directly computed by (10) and (11).

D. Actor-Critic Structure for Online SPUA and LS NN Approach

Actor-critic schemes [42] originated in the artificial intelligence literature in the context of RL. In the past three decades, actor-critic algorithms have received much attention (see [43] and [44] and references therein), and have been introduced to solve optimal control problems [11], [13].

In this section, we develop an actor-critic structure for the online SPUA (see Fig. 1) to solve the H_∞ state feedback control problem. This structure involves three learning units, a critic and two actors, interacting with each other and with the system during the course of the online SPUA. Two actors have tunable parameter vectors that parameterize a set of control policies and disturbance policies, respectively. They update their parameter vectors at each iterative step using the observations of the system state and the information obtained from the critic. Similarly, at each iterative step, the critic updates the approximation of cost function corresponding to the current control and disturbance policies of two actors.

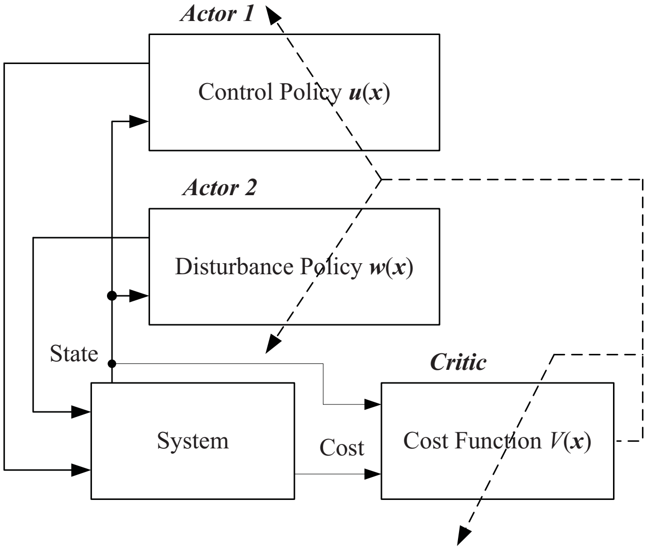


Fig. 1. Actor-critic structure for online SPUA.

If we use NNs to parameterize the cost function, control, and disturbance policies in the actor-critic structure, three NNs are needed. Here, we derive a simple method for implementation of this structure, in which only a single critic NN is required for the cost function, and then two actor NNs for the control and disturbance policies are updated accordingly.

Let $\Psi_N(x) = (\psi_1(x), \dots, \psi_N(x))^T$ be the activation functions, where N is the number of hidden-layer neurons. Then, the cost function $V^{(i+1)}(x)$ in (13) is approximated by

$$\hat{V}^{(i+1)}(x) = \left(c^{(i+1)}\right)^T \Psi_N(x) = \Psi_N^T(x)c^{(i+1)} \tag{28}$$

where $c^{(i+1)} = \left(c_1^{(i+1)}, \dots, c_N^{(i+1)}\right)^T$ is the weight vector. Thus, (13) can be written as

$$\begin{aligned}\Psi_N^T\Big(x(t)\Big)c^{(i+1)} = \int_t^{t+\Delta t} \bigg(\left\|h\Big(x(\tau)\Big)\right\|^2 + \left\|\hat{u}^{(i)}(\tau)\right\|_R^2 \\ -\gamma^2\left\|\hat{w}^{(i)}(\tau)\right\|^2\bigg)d\tau + \Psi_N^T\Big(x(t+\Delta t)\Big)c^{(i+1)}\end{aligned}$$

that means

$$\begin{aligned}&\Big(\Psi_N^T\left(x(t)\right) - \Psi_N^T\left(x(t+\Delta t)\right)\Big)c^{(i+1)} \\ &= \int_t^{t+\Delta t}\left(\|h\left(x(\tau)\right)\|^2 + \left\|\hat{u}^{(i)}(\tau)\right\|_R^2 - \gamma^2\left\|\hat{w}^{(i)}(\tau)\right\|^2\right)d\tau.\end{aligned} \tag{29}$$

Accordingly, the control and disturbance policies in (14) and (15) can be approximated by

$$\hat{u}^{(i+1)} = -\frac{1}{2}R^{-1}g^T\nabla\hat{V}^{(i+1)} = -\frac{1}{2}R^{-1}g^T\nabla\Psi_N^T c^{(i+1)} \tag{30}$$

$$\hat{w}^{(i+1)} = \frac{1}{2}\gamma^{-2}k^T\nabla\hat{V}^{(i+1)} = \frac{1}{2}\gamma^{-2}k^T\nabla\Psi_N^T c^{(i+1)} \tag{31}$$

where $\nabla\Psi_N(x) = ((\partial\psi_1/\partial x), \ldots, (\partial\psi_N/\partial x))^T$ is the Jacobian of Ψ_N.

Remark 5: Note that only one NN is needed in the online SPUA (Algorithm 1), i.e., the critic NN for approximating the cost function via (28). After the weight vector is computed via (29), the control and disturbance policies in (14) and (15) can be approximately updated by (30) and (31) accordingly. Therefore, the iteration from (13) to (15) in the online SPUA is converted to the weights iteration from (29) to (31).

It is noticed that the NN weight parameters $c^{(i+1)}$ have N unknown elements. Thus, in order to solve for $c^{(i+1)}$, at least N equations are required. Here, we construct $\bar{N}(\bar{N} \geq N)$ equations, and use an LS method to estimate $c^{(i+1)}$. In each time interval $[t, t+\Delta t]$, we collect $\bar{N}$ sample data along state trajectories, and construct the LS solution of the NN weights as follows:

$$c^{(i+1)} = (XX^T)^{-1}XY \tag{32}$$

where

$$X = \big[\,\Psi_N(x(t)) - \Psi_N(x(t+\delta t)) \cdots \Psi_N\big(x(t+(\bar{N}-1)\delta t)\big) - \Psi_N\big(x(t+\bar{N}\delta t)\big)\big]$$

$$Y = \Big[\,y(x(t), \hat{u}^{(i)}(t), \hat{w}^{(i)}(t)) \cdots y(x(t+(\bar{N}-1)\delta t), \hat{u}^{(i)}(t+(\bar{N}-1)\delta t), \hat{w}^{(i)}(t+(\bar{N}-1)\delta t))\Big]^T$$

with $\delta t = \Delta t/\bar{N}$ and

$$y(x(t+k\delta t), \hat{u}^{(i)}(t+k\delta t), \hat{w}^{(i)}(t+k\delta t)) = \int_{t+k\delta t}^{t+(k+1)\delta t}\left(\|h(x(\tau))\|^2 + \left\|\hat{u}^{(i)}(\tau)\right\|_R^2 - \gamma^2\left\|\hat{w}^{(i)}(\tau)\right\|^2\right)d\tau, \qquad k = 0, \ldots, N-1.$$

It is worth mentioning that the LS method (32) requires a nonsingular matrix XX^T. To attain the goal, we can inject probing noises into states or reset system states.

Based on the actor-critic structure and the above LS estimation of the NN weights, we develop an implementable NN-based online SPUA procedure as shown in Algorithm 2.

Remark 6: It should be pointed out that the word "simultaneous" in this paper and the word "synchronous" in [33] represent different meanings. The former emphasizes the same "iterative step," while the latter emphasizes the same "time instant. In this paper, the online SPUA updates control and disturbance policies at the same iterative step, while the algorithm in [33] updates control and disturbance policies at the different iterative steps.

Remark 7: It is worth emphasizing that different from the result in [11], which was used for solving HJB equation of nonlinear optimal control problem, the proposed online SPUA is developed for solving HJI equation of nonlinear H_∞ control problem. Moreover, there are two main differences between the approach in [34] and the proposed online SPUA. 1) The former is an offline approach that requires the system model, while the latter is an online one that does not need the knowledge of internal system dynamics and 2) The method in [34] brings two iterative loops, while the online SPUA involves only one loop.

Algorithm 2 NN-Based Online SPUA

Step 1: Select N activation functions $\Psi_N(x)$. Given initial weights $c^{(0)}$ such that $\hat{V}^{(0)} \in \mathbb{V}_0$, let

$$\hat{u}^{(0)} = -\frac{1}{2}R^{-1}g^T\nabla\Psi_N^Tc^{(0)},\ \hat{w}^{(0)} = \frac{1}{2}\gamma^{-2}k^T\nabla\Psi_N^Tc^{(0)},$$

and $i = 0$.

Step 2: With policies $\hat{u}^{(i)}$ and $\hat{w}^{(i)}$, collect $\bar{N}$ sample data along state trajectories in time interval $[i\Delta t, (i+1)\Delta t]$. Compute $c^{(i+1)}$ via (32) at time instant $(i+1)\Delta t$.

Step 3: Update the control and disturbance policies by (30) and (31) at time instant $(i+1)\Delta t$.

Step 4: Set $i = i+1$. If $\left\|c^{(i)} - c^{(i-1)}\right\| \leq \varepsilon$ (ε is a small positive real number), stop and use $V^{(i)}$ as the solution of the HJI equation (4), i.e., use $\hat{u}^{(i)}$ as the H_∞ controller, else, go back to step 2 and continue.

E. Convergence of LS NN-Based Online SPUA

In this section, we show that the LS NN-based online SPUA converges to the solution of HJI equation (4).

It is seen that the LS NN method derived in the above section is used for solving (13) for cost function $V^{(i+1)}(x)$. It follows from the proof of Theorem 1 that (13) is equal to (A5), which means that the LS NN approach is nothing but for solving (A5) mathematically.

We notice that (A5) is essentially the same as the Lyapunov equation in [45] from pure mathematical view, because both of them are first order linear PDEs of the same form. Some important theories have been established in computational mathematics community to solve these types of PDEs, such as the LS approach in [45] and Galerkin's method in [30] and [32]. Moreover, in [45], the convergence of LS NN approach for solving the first order, linear PDE was established, which provides the theoretical foundation of the online LS NN method in this paper. We can directly obtain the following Lemma by using the results in [45].

Lemma 6: For $i = 0, 1, 2, \ldots$, assume that the solution of equation (A5) $V^{(i+1)} \in H^{1,2}(\Omega)$, the NN activation functions $\psi_j \in H^{1,2}(\Omega)$, $j = 1, 2, \ldots, N$ are chosen such that, they are complete when $N \to \infty$, $V^{(i+1)}$ and $\nabla V^{(i+1)}$ can be uniformly approximated, and the set $\{\varphi_j(x_1, x_2) \overset{\Delta}{=} \psi_j(x_1) - \psi_j(x_2)\}_{j=1}^N$, $\forall x_1, x_2 \in \Omega$, $x_1 \neq x_2$ is linearly independent and complete. Then, for $i = 0, 1, 2$

$$\sup_{x\in\Omega}\left|\hat{V}^{(i+1)}(x) - V^{(i+1)}(x)\right| \to 0$$

$$\sup_{x\in\Omega}\left|\nabla\hat{V}^{(i+1)}(x) - \nabla V^{(i+1)}(x)\right| \to 0$$

$$\sup_{x\in\Omega}\left|\hat{u}^{(i+1)}(x) - u^{(i+1)}(x)\right| \to 0$$

$$\sup_{x\in\Omega}\left|\hat{w}^{(i+1)}(x) - w^{(i+1)}(x)\right| \to 0.$$

Proof: In order to use the results in [45], we first show the set$\{\nabla\psi_j^T(f + gu^{(i)} + kw^{(i)})\}$is linearly independent by contradiction. Suppose this is not true, then there exists a nonzero vector $\widehat{c} \triangleq \left[\widehat{c}_1 \ldots \widehat{c}_N\right] \in \mathbb{R}^N$ such that

$$\sum_{j=1}^{N} \widehat{c}_j \nabla \psi_j^T (f + gu^{(i)} + kw^{(i)}) = 0$$

which implies that for $\forall x(t) \in \Omega$

$$\int_t^{t+\Delta t} \sum_{j=1}^{N} \widehat{c}_j \nabla \psi_j^T (f + gu^{(i)} + kw^{(i)}) d\tau = \sum_{j=1}^{N} \widehat{c}_j \left(\psi_j(x(t+\Delta t)) - \psi_j(x(t))\right) = 0$$

that means

$$\sum_{j=1}^{N} \widehat{c}_j \varphi_j(x(t+\Delta t), x(t)) = 0.$$

This contradicts the linear independence of $\{\varphi_j\}_{j=1}^N$. Thus, the set$\{\nabla \psi_j^T (f + gu^{(i)} + kw^{(i)})\}$ is linearly independent.

Then, the first three items of Lemma 6 can be proved by following the same procedure used in Theorem 2 and Corollary 2 of [45]. The result $\sup_{x\in\Omega} \left|\hat{w}^{(i+1)}(x) - w^{(i+1)}(x)\right| \to 0$ can also be proved by arguments similar to $\sup_{x\in\Omega} \left|\hat{u}^{(i+1)}(x) - u^{(i+1)}(x)\right| \to 0$.

Lemma 6 shows the LS NN approach can achieve the uniform approximation for the solution of (A5).

Theorem 2: If the conditions in Lemma 6 hold, then, for $\forall \varsigma > 0$, $\exists i_0, N_0$, when $i \geq i_0$ and $N \geq N_0$, we have

$$\sup_{x\in\Omega} \left|\hat{V}^{(i)}(x) - V^*(x)\right| < \varsigma$$
$$\sup_{x\in\Omega} \left|\hat{u}^{(i)}(x) - u^*(x)\right| < \varsigma$$
$$\sup_{x\in\Omega} \left|\hat{w}^{(i)}(x) - w^*(x)\right| < \varsigma.$$

Proof: The first two items can be proved by following the same procedure used in the proofs of Theorems 3 and 4 of [45]. The result $\sup_{x\in\Omega} \left|\hat{w}^{(i)}(x) - w^*(x)\right| \to 0$ can also be proved in a similar way of $\sup_{x\in\Omega} \left|\hat{u}^{(i)}(x) - u^*(x)\right| \to 0$. ■

Theorem 2 demonstrates the uniform convergence of the proposed online SPUA with LS NN approximation.

Remark 8: Observe that the convergence proof of the proposed LS NN approach in this paper is almost the same as that in [45]. The reason is that both the proposed LS NN approach in this paper and one in [45] are developed for solving a first order linear PDE. However, the final goal of [45] is to solve a HJB equation of nonlinear optimal control problem, while our aim is to solve a HJI equation in nonlinear H_∞ control problem. On the other hand, the method in [45] is an offline one that requires the system model, while our method is an online one without requiring the knowledge of internal system dynamics.

IV. Simulation Studies

In this section, we present simulation studies on two examples to illustrate the effectiveness of the developed NN-based online SPUA.

A. Simulations on Linear System

The first example considers the following F-16 aircraft plant that studied in [47] and [48]

$$\dot{x} = Ax + B_1 w + B_2 u, \quad z = Cx \tag{33}$$

where $C = I$ and

$$A = \begin{bmatrix} -1.01887 & 0.90506 & -0.00215 \\ 0.82225 & -1.07741 & -0.17555 \\ 0 & 0 & -1 \end{bmatrix}, \quad B_1 = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}$$
$$B_2 = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}$$

the system state vector is $x = \left[\alpha \ q \ \delta_e\right]^T$, α denotes the angle of attack, q is the rate, and δ_e is the elevator deflection angle. The control input u is the elevator actuator voltage and the disturbance w is wind gusts on angle of attack. Select $R = I$ and $\gamma = 5$. Letting $V^*(x) = x^T Px$, the HJI equation (4) for linear system (33) is the following H_∞ ARE:

$$A^T P + PA + C^T C + \gamma^{-2} PB_1 B_1^T P - PB_2 R^{-1} B_2^T P = 0 \tag{34}$$

and the corresponding H_∞ control law (5) is

$$u^*(x) = -R^{-1} B_2^T Px. \tag{35}$$

Solving the ARE (34) with the MATLAB command CARE, we obtain

$$P = \begin{bmatrix} 1.6573 & 1.3954 & -0.1661 \\ 1.3954 & 1.6573 & -0.1804 \\ -0.1661 & -0.1804 & 0.4371 \end{bmatrix}. \tag{36}$$

Here, we use the proposed NN-based online SPUA to solve the H_∞ control problem of system (33). Select six polynomials as activation functions as follows:

$$\Psi_N(x) = \left[x_1^2 \ x_1x_2 \ x_1x_3 \ x_2^2 \ x_2x_3 \ x_3^2\right]^T$$

thus, the true values of the NN weights c are

$$c = \left[P_{11} \ 2P_{12} \ 2P_{13} \ P_{22} \ 2P_{23} \ P_{33}\right]^T = \left[1.6573 \ 2.7908 \ -0.3322 \ 1.6573 \ -0.3608 \ 0.4370\right]^T. \tag{37}$$

Select the value of stop criterion $\varepsilon = 10^{-7}$, initial state $x(0) = \left[1 \ 1 \ 1\right]^T$, the initial NN weights $c^{(0)} = 0$, and sampling interval $\delta t = 0.05(s)$. In each iterative step, after collecting 10 (i.e., $\bar{N} = 10$) system state measurements, the LS method (32) is used to update NN weights, that is, the NN weights are updated every $0.5(s)$ (i.e., $\Delta t = 0.5(s)$). After each update, we reset the system state as initial state $\mathbf{x}_0$. Fig. 2 shows the weights $c^{(i)}$ in each iterative step, where we can observe that the NN weights converge to the true values in (37) at $t_0 = 2.5(s)$. Then the solution of HJI equation is computed via (28) and the corresponding H_∞ controller is obtained by (30). Select a disturbance signal as

$$w(t) = \begin{cases} 8e^{-(t-t_0)} \cos(t - t_0), & t \geq t_0 \\ 0, & t < t_0 \end{cases} \tag{38}$$

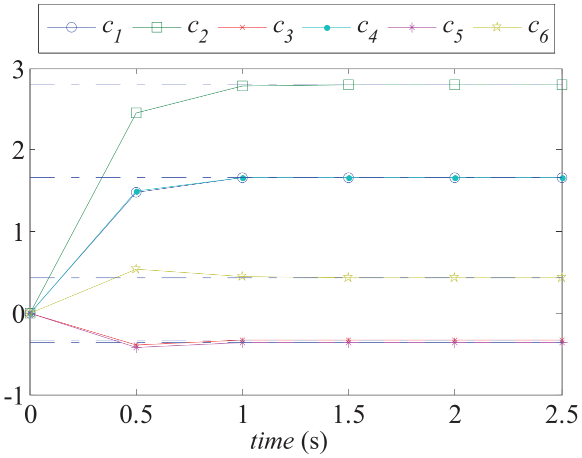
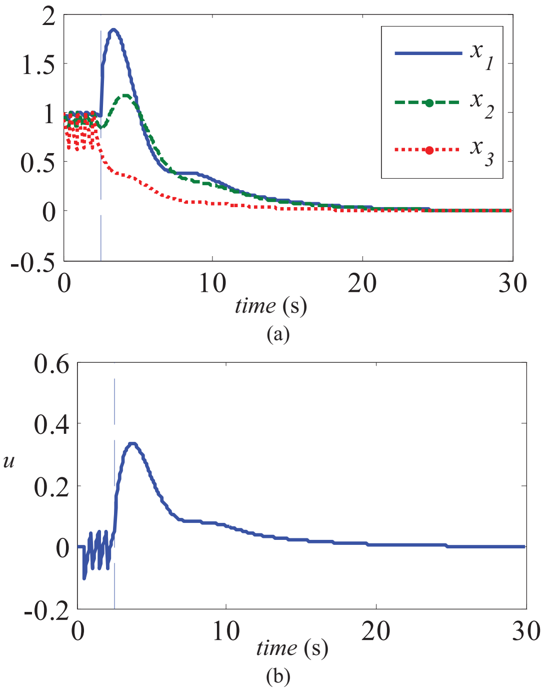


Fig. 2. NN weights for the first example with $\Delta t = 0.5(s)$.



(a)

(b)

Fig. 3. For the first example (a) closed-loop state trajectories and (b) control action $u(t)$.

and use the resulting H_∞ controller for closed-loop system simulations. Fig. 3 shows the closed-loop state trajectories and control action $u(t)$. The trajectories at the first 2.5 s are corresponding to a phrase in which the online SPUA is applied to learn the NN weights.

In order to test the influence of different Δt on the performance of the online SPUA, we run the online SPUA on Example 1 again under the same above parameters except by setting $\Delta t = 1(s)$ (i.e., $\delta t = 0.1(s)$). Fig. 4 gives the NN weights in each iterative step. It is noticed that the weights are convergent at $t_0 = 5(s)$, which is doubled compared with Fig. 2. However, it is also found that the online SPUA is convergent at iterative step 5 (i.e., $i = 5$), which is the same as the results obtained by setting $\Delta t = 0.5(s)$. This means that the change of Δt have great effect on the time of convergence, but little on the iterative steps of convergence.

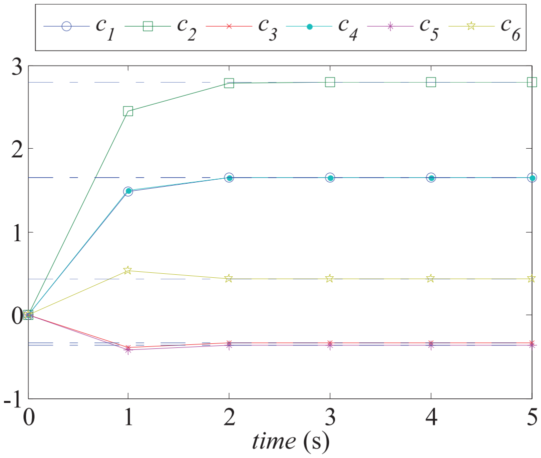


Fig. 4. NN weights for the first example with $\Delta t = 1(s)$.

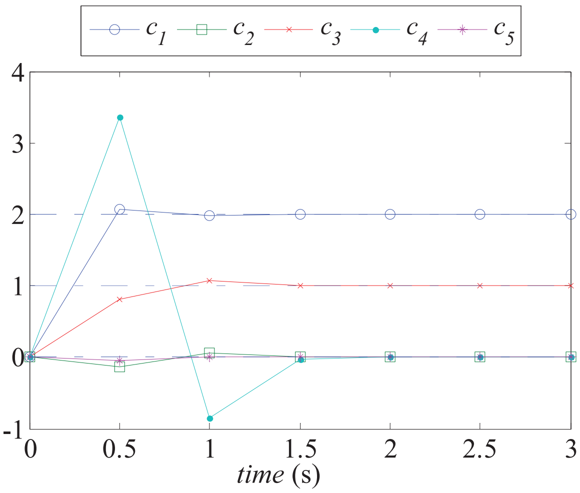


Fig. 5. NN weights for the second example.

B. Simulations on Nonlinear System

The second example is constructed by using the converse HJB approach [49]. The system model is given as follows:

$$\dot{x} = \begin{bmatrix} -0.25x_1 \\ 0.5x_1^2x_2 - 0.5\gamma^{-2}x_2^3 + 0.5x_2 \end{bmatrix} + \begin{bmatrix} 0 \\ x_1 \end{bmatrix} w + \begin{bmatrix} 0 \\ x_2 \end{bmatrix} u$$

$$z = x.$$

With the choice of $\gamma = 2$, the solution of the associated HJI equation is $V^*(x) = 2x_1^2 + x_2^2$.

Select $R = I$, $x(0) = [\,0.4\ 0.5\,]^T$, $\varepsilon = 10^{-7}$, NN activation functions $\Psi_N(x) = [\,x_1^2\ x_1x_2\ x_2^2\ \ x_1^4\ x_2^4\,]^T$, and the initial NN weights $c^{(0)} = 0$. The parameters of states sampling are the same as Example 1. After each update, we reset the system state as initial state x_0. By the proposed NN-based online SPUA, the simulation results are shown in Figs. 5 and 6. Fig. 5 indicates the weights $c^{(i)}$ in each iterative step, where it can be observed that the NN weights converge to the true weights (i.e., $\begin{bmatrix} 2\ 0\ 1\ \ 0\ 0 \end{bmatrix}^T$) at $t_0 = 3(s)$. By the resulting NN weights at instant $t_0 = 3(s)$, we can obtain the solution of HJI equation (4) by (28) and the corresponding H_∞ controller by (30). Select a disturbance signal as in (38) with $t_0 = 3(s)$

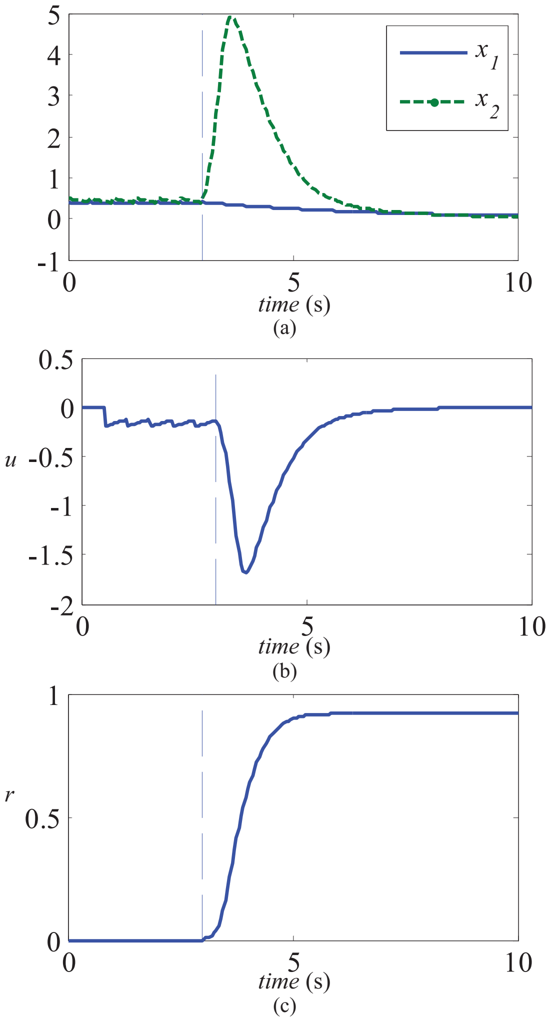


Fig. 6. For the second example (a) closed-loop state trajectories, (b) control action $u(t)$, and (c) evolution of $r(t)$.

and define the function $r(t)$ as

$$r(t) \triangleq \frac{\int_{t_0}^{t} \left(\|z(\tau)\|^2 + \|u(\tau)\|_R^2\right) d\tau}{\int_{t_0}^{t} \|w(\tau)\|^2 d\tau}. \tag{39}$$

Applying the resulting H_∞ controller to the system, Fig. 6 shows the closed-loop state trajectories, control action $u(t)$, and evolution of $r(t)$. The trajectories at the first 3 s in Fig. 6 are corresponding to a learning phrase, in which the online SPUA is used to learn the NN weights. It can also be seen from Fig. 6 that the closed-loop system is asymptotically stable, and the $r(t)$ converges to 0.9232, which satisfies the L_2-gain requirement (i.e., $r(t) < \gamma^2 = 4$) when $t \to \infty$.

V. Conclusion

In this paper, a NN-based online SPUA has been developed to solve the HJI equation of H_∞ state feedback control problem for nonlinear systems. The H_∞control problem was viewed as a zero-sum game, where the control is a minimizing player and the disturbance is a maximizing one. By updating two players' policies simultaneously, an online SPUA was proposed to learn the solution of the HJI equation. The convergence of the online SPUA was established by showing that it is mathematically equivalent to Newton's method for finding a fixed point in a Banach space. Moreover, for implementation purpose, we presented an actor-critic structure, in which two actors and one critic were involved. To simplify this structure, only a single critic NN was employed for approximating the cost function, and then two actor NNs for the control and disturbance policies were updated correspondingly. Furthermore, an LS method was given to estimate the NN weight parameters, and its convergence was proved. Finally, through the simulation studies on two examples, the achieved results showed that the proposed NN-based online SPUA is effective.

Appendix

Proof of Lemma 3

For $\forall V \in \mathbb{V}$ and $W \in \tilde{\mathbb{V}} \subset \mathbb{V}$, where $\tilde{\mathbb{V}}$ is a neighborhood of V, we have

$$\begin{aligned}
&G(V+sW) - G(V)\\
&= (\nabla(V+sW))^T f + h^T h - (\nabla(V+sW))^T gR^{-1}\\
&\quad \times g^T \nabla(V+sW) + \frac{1}{4\gamma^2}(\nabla(V+sW))^T kk^T \nabla(V+sW)\\
&\quad -\Big((\nabla V)^T f + h^T h - (\nabla V)^T gR^{-1}g^T \nabla V\\
&\qquad + \frac{1}{4\gamma^2}(\nabla V)^T kk^T \nabla V\Big)\\
&= s(\nabla W)^T f - \frac{s}{4}(\nabla W)^T gR^{-1}g^T \nabla V\\
&\quad - \frac{s}{4}(\nabla V)^T gR^{-1}g^T \nabla W\\
&\quad - \frac{s^2}{4}(\nabla W)^T gR^{-1}g^T \nabla W + \frac{s}{4\gamma^2}(\nabla W)^T kk^T \nabla V\\
&\quad + \frac{s}{4\gamma^2}(\nabla V)^T kk^T \nabla W + \frac{s^2}{4\gamma^2}(\nabla W)^T kk^T \nabla W.
\end{aligned}$$

Thus, the Gâteaux differential at V is

$$\begin{aligned}
L(W) &= \lim_{s\to 0} \frac{G(V+sW) - G(V)}{s}\\
&= (\nabla W)^T f - \frac{1}{4}(\nabla W)^T gR^{-1}g^T \nabla V - \frac{1}{4}(\nabla V)^T\\
&\quad \times gR^{-1}g^T \nabla W + \frac{1}{4\gamma^2}(\nabla W)^T kk^T \nabla V\\
&\quad + \frac{1}{4\gamma^2}(\nabla V)^T kk^T \nabla W.
\end{aligned} \tag{A1}$$

Next, we will prove that the map $L = G'(V)$ is continuous. For $\forall W_0 \in \mathbb{V}$, it is immediate that

$$\begin{aligned}
L(W) - L(W_0) = (\nabla(W-W_0))^T f &- \frac{1}{4}(\nabla(W-W_0))^T\\
&\times gR^{-1}g^T \nabla V\\
&- \frac{1}{4}(\nabla V)^T gR^{-1}g^T \nabla(W-W_0)\\
&+ \frac{1}{4\gamma^2}(\nabla(W-W_0))^T kk^T \frac{\partial V}{\partial x}\\
&+ \frac{1}{4\gamma^2}(\nabla V)^T kk^T \nabla(W-W_0).
\end{aligned}$$

Then, we have

$$\begin{aligned}
&\|L(W)-L(W_0)\|_\Omega \\
&\le \left\|(\nabla(W-W_0))^T f\right\|_\Omega + \left\|\frac{1}{4}(\nabla(W-W_0))^T\right. \\
&\quad \times \left. gR^{-1}g^T\nabla V\right\|_\Omega + \left\|\frac{1}{4}(\nabla V)^T gR^{-1}g^T\nabla(W-W_0)\right\|_\Omega \\
&\quad + \left\|\frac{1}{4\gamma^2}(\nabla(W-W_0))^T kk^T\nabla V\right\|_\Omega \\
&\quad + \left\|\frac{1}{4\gamma^2}(\nabla V)^T kk^T\nabla(W-W_0)\right\|_\Omega \\
&= \left(\|f\|_\Omega + \left\|\frac{1}{2}gR^{-1}g^T\nabla V\right\|_\Omega + \left\|\frac{1}{2\gamma^2}kk^T\nabla V\right\|_\Omega\right) \\
&\quad \times \|\nabla(W-W_0)\|_\Omega \\
&\le \left(\|f\|_\Omega + \left\|\frac{1}{2}gR^{-1}g^T\nabla V\right\|_\Omega + \left\|\frac{1}{2\gamma^2}kk^T\nabla V\right\|_\Omega\right) \\
&\quad \times m_1\|W-W_0\|_\Omega
\end{aligned} \tag{A2}$$

where $m_1 > 0$. Let

$$M = m_1\left(\|f\|_\Omega + \left\|\frac{1}{2}gR^{-1}g^T\nabla V\right\|_\Omega + \left\|\frac{1}{2\gamma^2}kk^T\nabla V\right\|_\Omega\right).$$

Then, for $\forall\varepsilon > 0$, there exists a $\delta = \varepsilon/M$ such that

$$\|L(W)-\mathcal{L}(W_0)\|_\Omega \le M\|W-W_0\|_\Omega < \varepsilon \tag{A3}$$

when $\|W-W_0\|_\Omega < \delta$. This means $L = G'(V)$ is continuous on $\tilde{\mathbb{V}}$, thus according to Lemma 2, $L(W) = G'(V)W$ [i.e., (A1)] is the Fréchet differential, and $L = G'(V)$ is the Fréchet derivative at V. ■

Proof of Theorem 1: We will give the proof with two steps.

1) On the one hand, with policies $u^{(i)}$ and $w^{(i)}$, the system state $x(t)$ satisfies

$$\dot{x}(t) = f(x) + g(x)u^{(i)}(t) + k(x)w^{(i)}(t). \tag{A4}$$

Let $V^{(i+1)}$ be a solution of the following equations:

$$\begin{aligned}
&\left(\nabla V^{(i+1)}\right)^T\left(f + gu^{(i)} + kw^{(i)}\right) + \|h\|^2 + \left\|u^{(i)}\right\|_R^2 \\
&\qquad -\gamma^2\left\|w^{(i)}\right\|^2 = 0.
\end{aligned} \tag{A5}$$

Using (A4), (A5) can be rewritten as

$$\frac{d}{dt}V^{(i+1)}(x(t)) = -\left(\|h\|^2 + \left\|u^{(i)}\right\|_R^2 - \gamma^2\left\|w^{(i)}\right\|^2\right). \tag{A6}$$

Integrating (A6) from t to $t+\Delta t$ yields

$$\begin{aligned}
&V^{(i+1)}(x(t+\Delta t)) - V^{(i+1)}(x(t)) = -\int_t^{t+\Delta t} \\
&\quad \times \left(\|h(x(\tau))\|^2 + \left\|u^{(i)}(\tau)\right\|_R^2 - \gamma^2\left\|w^{(i)}(\tau)\right\|^2\right)d\tau
\end{aligned} \tag{A7}$$

which implies that the solution $V^{(i+1)}$ satisfies (13). Obviously, (13) can be rewritten as

$$\begin{aligned}
V^{(i+1)}\Big(x(t)\Big) = \int_t^\infty &\left(\left\|h\Big(x(\tau)\Big)\right\|^2 + \left\|u^{(i)}(\tau)\right\|_R^2\right. \\
&\left. -\gamma^2\left\|w^{(i)}(\tau)\right\|^2\right)d\tau.
\end{aligned} \tag{A8}$$

Calculating the derivative of (A8), yields

$$\begin{aligned}
\dot{V}^{(i+1)}(x(t)) &= \left(\nabla V^{(i+1)}\right)^T\dot{x} \\
&= -\left(\|h(x(t))\|^2 + \left\|u^{(i)}(t)\right\|_R^2 - \gamma^2\left\|w^{(i)}(t)\right\|^2\right).
\end{aligned} \tag{A9}$$

In the following, we prove that $V^{(i+1)}$ is the unique solution of (13) by contradiction. Assume $\overset{\frown}{V}^{(i+1)}$ is another solution of (13), that is

$$\begin{aligned}
\overset{\frown}{V}^{(i+1)}(x(t)) = \int_t^\infty &\left(\|h(x(\tau))\|^2 + \left\|u^{(i)}(\tau)\right\|_R^2\right. \\
&\left. -\gamma^2\left\|w^{(i)}(\tau)\right\|^2\right)d\tau.
\end{aligned} \tag{A10}$$

Then the derivative of (A10) can be calculated as

$$\begin{aligned}
\dot{\overset{\frown}{V}}^{(i+1)}(x(t)) &= \left(\nabla\overset{\frown}{V}^{(i+1)}\right)^T\dot{x} \\
&= -\left(\|h(x(t))\|^2 + \left\|u^{(i)}(t)\right\|_R^2 - \gamma^2\left\|w^{(i)}(t)\right\|^2\right).
\end{aligned} \tag{A11}$$

It follows from (A9) and (A11) that

$$\left(\nabla\left(V^{(i+1)} - \overset{\frown}{V}^{(i+1)}\right)\right)^T\dot{x} = 0 \tag{A12}$$

which must hold for any $x \in \Omega$. Thus, we have

$$\frac{d}{dt}\left(V^{(i+1)} - \overset{\frown}{V}^{(i+1)}\right) = 0. \tag{A13}$$

This means $\overset{\frown}{V}^{(i+1)}(x) = V^{(i+1)}(x) - d$, where$d$ is a constant. Due to $\overset{\frown}{V}^{(i+1)}(0) = V^{(i+1)}(0) = 0$ and $\overset{\frown}{V}^{(i+1)}(0) = V^{(i+1)}(0) - d$, then $d = 0$. Thus, $\overset{\frown}{V}^{(i+1)}(x) = V^{(i+1)}(x)$. Therefore, the (13) is equal to (A5).

2) On the other hand, it follows from (16) and (20) that:

$$V^{(i+1)} = TV^{(i)} = V^{(i)} - \left(G'(V^{(i)})\right)^{-1}G(V^{(i)})$$

which can be rewritten as

$$G'(V^{(i)})V^{(i+1)} = G'(V^{(i)})V^{(i)} - G(V^{(i)}). \tag{A14}$$

From (14), (15), and (19), we have

$$\begin{aligned}
&G'(V^{(i)})V^{(i+1)} = \left(\nabla V^{(i+1)}\right)^T f - \frac{1}{4}\left(\nabla V^{(i+1)}\right)^T \\
&\quad \times gR^{-1}g^T\nabla V^{(i)} \\
&\quad - \frac{1}{4}\left(\nabla V^{(i)}\right)^T gR^{-1}g^T\nabla V^{(i+1)} + \frac{1}{4\gamma^2}\left(\nabla V^{(i+1)}\right)^T \\
&\quad \times kk^T\nabla V^{(i)} + \frac{1}{4\gamma^2}\left(\nabla V^{(i)}\right)^T kk^T\nabla V^{(i+1)} \\
&= \left(\nabla V^{(i+1)}\right)^T f - \frac{1}{4}\left(\nabla V^{(i+1)}\right)^T gR^{-1}g^T\nabla V^{(i)} \\
&\quad - \left(\frac{1}{4}\left(\nabla V^{(i)}\right)^T gR^{-1}g^T\nabla V^{(i+1)}\right)^T + \frac{1}{4\gamma^2}\left(\nabla V^{(i+1)}\right)^T \\
&\quad \times kk^T\nabla V^{(i)} + \frac{1}{4\gamma^2}\left(\left(\nabla V^{(i)}\right)^T kk^T\nabla V^{(i+1)}\right)^T \\
&= \left(\nabla V^{(i+1)}\right)^T\left(f - \frac{1}{2}gR^{-1}g^T\nabla V^{(i)} + \frac{1}{2\gamma^2}kk^T\nabla V^{(i)}\right)
\end{aligned}$$

$$= \left(\nabla V^{(i+1)}\right)^T \left(f + gu^{(i)} + kw^{(i)}\right) \tag{A15}$$

$$\begin{aligned} G'(V^{(i)})V^{(i)} &= \left(\nabla V^{(i)}\right)^T f - \frac{1}{4}\left(\nabla V^{(i)}\right)^T gR^{-1}g^T\nabla V^{(i)} \\ &\quad - \frac{1}{4}\left(\nabla V^{(i)}\right)^T gR^{-1}g^T\nabla V^{(i)} + \frac{1}{4\gamma^2}\left(\nabla V^{(i)}\right)^T \\ &\quad \times kk^T\nabla V^{(i)} + \frac{1}{4\gamma^2}\left(\nabla V^{(i)}\right)^T kk^T\nabla V^{(i)} \end{aligned}$$

$$= \left(\nabla V^{(i)}\right)^T f - 2\left\|u^{(i)}\right\|_R^2 + 2\gamma^2\left\|w^{(i)}\right\|^2 \tag{A16}$$

$$\begin{aligned} G(V^{(i)}) &= \left(\nabla V^{(i)}\right)^T f + h^T h - \frac{1}{4}\left(\nabla V^{(i)}\right)^T \\ &\quad \times gR^{-1}g^T\nabla V^{(i)} + \frac{1}{4\gamma^2}\left(\nabla V^{(i)}\right)^T kk^T\nabla V^{(i)} \end{aligned}$$

$$= \left(\nabla V^{(i)}\right)^T f + \|h\|^2 - \left\|u^{(i)}\right\|_R^2 + \gamma^2\left\|w^{(i)}\right\|^2. \tag{A17}$$

Substituting (A15)–(A17) into (A14) gives

$$\begin{aligned} &\left(\nabla V^{(i+1)}\right)^T \left(f + gu^{(i)} + kw^{(i)}\right) \\ &\quad = \left(\nabla V^{(i)}\right)^T f - 2\left\|u^{(i)}\right\|_R^2 + 2\gamma^2\left\|w^{(i)}\right\|^2 \\ &\qquad - \left(\left(\nabla V^{(i)}\right)^T f + \|h\|^2 - \left\|u^{(i)}\right\|_R^2 + \gamma^2\left\|w^{(i)}\right\|^2\right) \end{aligned}$$

that means

$$\begin{aligned} \left(\nabla V^{(i+1)}\right)^T \left(f + gu^{(i)} + kw^{(i)}\right) + \|h\|^2 + \left\|u^{(i)}\right\|_R^2 \\ -\gamma^2\left\|w^{(i)}\right\|^2 = 0. \end{aligned}$$

This means that the (20) in Newton's iteration is also equal to (A5). This completes the proof.

Acknowledgment

The authors would like to thank the Associate Editor and the anonymous reviewers for their helpful comments and suggestions, which have improved the presentation of this paper.

References

[1] T. Başar and P. Bernhard, *H^∞ Optimal Control and Related Minimax Design Problems: A Dynamic Game Approach*, 2nd ed. Boston, MA: Birkhäuser, 1995.

[2] A. J. van der Schaft, *L_2-Gain and Passivity Techniques in Nonlinear Control*. Berlin, Germany: Springer-Verlag, 1996.

[3] K. Zhou, J. C. Doyle, and K. Glover, *Robust and Optimal Control*. Upper Saddle River, NJ: Prentice-Hall, 1996.

[4] A. Isidori and W. Kang, "H_∞ control via measurement feedback for general nonlinear systems," *IEEE Trans. Autom. Control*, vol. 40, no. 3, pp. 466–472, Mar. 1995.

[5] G. Bianchini, R. Genesio, A. Parenti, and A. Tesi, "Global H_∞ controllers for a class of nonlinear systems," *IEEE Trans. Autom. Control*, vol. 49, no. 2, pp. 244–249, Feb. 2004.

[6] A. J. van der Schaft, "L_2-gain analysis of nonlinear systems and nonlinear state-feedback H_∞ control," *IEEE Trans. Autom. Control*, vol. 37, no. 6, pp. 770–784, Jun. 1992.

[7] A. Al-Tamimi, F. L. Lewis, and M. Abu-Khalaf, "Discrete-time nonlinear HJB solution using approximate dynamic programming: Convergence proof," *IEEE Trans. Syst., Man, Cybern., B, Cybern.*, vol. 38, no. 4, pp. 943–949, Aug. 2008.

[8] H. Zhang, Q. Wei, and Y. Luo, "A novel infinite-time optimal tracking control scheme for a class of discrete-time nonlinear systems via the greedy HDP iteration algorithm," *IEEE Trans. Syst., Man, Cybern., B, Cybern.*, vol. 38, no. 4, pp. 937–942, Aug. 2008.

[9] Q. M. Yang, J. B. Vance, and S. Jagannathan, "Control of nonaffine nonlinear discrete-time systems using reinforcement-learning-based linearly parameterized neural networks," *IEEE Trans. Syst., Man, Cybern., B, Cybern.*, vol. 38, no. 4, pp. 994–1001, Aug. 2008.

[10] D. Vrabie, O. Pastravanu, M. Abu-Khalaf, and F. L. Lewis, "Adaptive optimal control for continuous-time linear systems based on policy iteration," *Automatica*, vol. 45, no. 2, pp. 477–484, 2009.

[11] D. Vrabie and F. L. Lewis, "Neural network approach to continuous-time direct adaptive optimal control for partially unknown nonlinear systems," *Neural Netw.*, vol. 22, no. 3, pp. 237–246, 2009.

[12] H. Zhang, Y. Luo, and D. Liu, "Neural-network-based near-optimal control for a class of discrete-time affine nonlinear systems with control constraints," *IEEE Trans. Neural Netw.*, vol. 20, no. 9, pp. 1490–1503, Sep. 2009.

[13] K. G. Vamvoudakis and F. L. Lewis, "Online actor-critic algorithm to solve the continuous-time infinite horizon optimal control problem," *Automatica*, vol. 46, no. 5, pp. 878–888, 2010.

[14] F. Wang, N. Jin, D. Liu, and Q. Wei, "Adaptive dynamic programming for finite-horizon optimal control of discrete-time nonlinear systems with ε-error bound," *IEEE Trans. Neural Netw.*, vol. 22, no. 1, pp. 24–36, Jan. 2011.

[15] T. Dierks and S. Jagannathan, "Online optimal control of affine nonlinear discrete-time systems with unknown internal dynamics by using time-based policy update," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 23, no. 7, pp. 1118–1129, Jul. 2012.

[16] H. Zhang, L. Cui, X. Zhang, and Y. Luo, "Data-driven robust approximate optimal tracking control for unknown general nonlinear systems using adaptive dynamic programming method," *IEEE Trans. Neural Netw.*, vol. 22, no. 12, pp. 2226–2236, Dec. 2011.

[17] Y. Jiang and Z. P. Jiang, "Approximate dynamic programming for optimal stationary control with control-dependent noise," *IEEE Trans. Neural Netw.*, vol. 22, no. 12, pp. 2392–2398, Dec. 2011.

[18] H. Zhang, R. Song, Q. Wei, and T. Zhang, "Optimal tracking control for a class of nonlinear discrete-time systems with time delays based on heuristic dynamic programming," *IEEE Trans. Neural Netw.*, vol. 22, no. 12, pp. 1851–1862, Dec. 2011.

[19] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*. Cambridge, MA: MIT Press, 1998.

[20] F. L. Lewis and D. Vrabie, "Reinforcement learning and adaptive dynamic programming for feedback control," *IEEE Circuits Syst. Mag.*, vol. 9, no. 3, pp. 32–50, Sep. 2009.

[21] F. Wang, H. Zhang, and D. Liu, "Adaptive dynamic programming: An introduction," *IEEE Comput. Intell. Mag.*, vol. 4, no. 2, pp. 39–47, May 2009.

[22] W. B. Powell, *Approximate Dynamic Programming: Solving the Curses of Dimensionality*. Hoboken, NJ: Wiley, 2007.

[23] D. P. Bertsekas and J. N. Tsitsiklis, *Neuro-Dynamic Programming*. Belmont, MA: Athena Scientific, 1996.

[24] X. Xu, C. Liu, S. X. Yang, and D. Hu, "Hierarchical approximate policy iteration with binary-tree state space decomposition," *IEEE Trans. Neural Netw.*, vol. 22, no. 12, pp. 1863–1877, Dec. 2011.

[25] M. Fairbank, E. Alonso, and D. Prokhorov, "Simple and fast calculation of the second-order gradients for globalized dual heuristic dynamic programming in neural networks," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 23, no. 10, pp. 1671–1676, Oct. 2012.

[26] Y. Feng, B. Anderson, and M. Rotkowitz, "A game theoretic algorithm to compute local stabilizing solutions to HJBI equations in nonlinear H_∞ control," *Automatica*, vol. 45, no. 4, pp. 881–888, 2009.

[27] A. Lanzon, Y. Feng, B. D. O. Anderson, and M. Rotkowitz, "Computing the positive stabilizing solution to algebraic Riccati equations with an indefinite quadratic term via a recursive method," *IEEE Trans. Autom. Control*, vol. 53, no. 10, pp. 2280–2291, Nov. 2008.

[28] J. Huang and C. Lin, "Numerical approach to computing nonlinear H_∞ control laws," *AIAA J. Guidance, Control, Dynamics*, vol. 18, no. 5, pp. 989–994, 1995.

[29] G. N. Saridis and C. G. Lee, "An approximation theory of optimal control for trainable manipulators," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 9, no. 3, pp. 152–159, Mar. 1979.

[30] R. Beard, G. N. Saridis, and J. Wen, "Galerkin approximations of the generalized Hamilton-Jacobi-Bellman equation," *Automatica*, vol. 33, no. 12, pp. 2159–2177, 1997.

[31] R. Beard, G. N. Saridis, and J. Wen, "Approximate solutions to the time-invariant Hamilton–Jacobi–Bellman equation," *J. Optim. Theory Appl.*, vol. 96, no. 3, pp. 589–626, 1998.

[32] R. W. Beard and T. W. Mclain, "Successive Galerkin approximation algorithms for nonlinear optimal and robust control," *Int. J. Control*, vol. 71, no. 5, pp. 717–743, 1998.

[33] K. G. Vamvoudakis and F. L. Lewis, "Online solution of nonlinear two-player zero-sum games using synchronous policy iteration," *Int. J. Robust Nonlinear Control*, vol. 22, no. 13, pp. 1460–1483, 2011.

[34] M. Abu-Khalaf, F. L. Lewis, and J. Huang, "Policy iterations on the Hamilton–Jacobi–Isaacs equation for H_∞ state feedback control with input saturation," *IEEE Trans. Autom. Control*, vol. 51, no. 12, pp. 1989–1995, Dec. 2006.

[35] M. Abu-Khalaf, F. L. Lewis, and J. Huang, "Neurodynamic programming and zero-sum games for constrained control systems," *IEEE Trans. Neural Netw.*, vol. 19, no. 7, pp. 1243–1252, Jul. 2008.

[36] M. Abu-Khalaf, J. Huang, and F. L. Lewis, *Nonlinear H2/H-Infinity Constrained Feedback Control: A Practical Design Approach Using Neural Networks*. New York: Springer-Verlag, 2006.

[37] D. Vrabie and F. L. Lewis, "Adaptive dynamic programming for online solution of a zero-sum differential game," *J. Control Theory Appl.*, vol. 9, no. 3, pp. 353–360, 2011.

[38] E. Zeidler, *Nonlinear Functional Analysis: Fixed Point Theorems*, vol. 1. New York: Springer-Verlag, 1985.

[39] L. Kantorovitch, "The method of successive approximation for functional equations," *Acta Math.*, vol. 71, no. 1, pp. 63–97, 1939.

[40] R. A. Tapia, "The Kantorovich theorem for Newton's method," *Amer. Math. Monthly*, vol. 78, no. 4, pp. 389–392, 1971.

[41] L. B. Rall, "A note on the convergence of Newton's method," *SIAM J. Numer. Anal.*, vol. 11, no. 1, pp. 34–36, 1974.

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

[43] V. Konda, "On actor-critic algorithms," Ph.D dissertation, Dept. Electr. Eng. & Comput. Sci., Massachusetts Inst. Technology, Cambridge, 2002.

[44] V. Konda and J. N. Tsitsiklis, "On actor-critic algorithms," *SIAM J. Control Optim.*, vol. 42, no. 4, pp. 1143–1166, 2003.

[45] M. Abu-Khalaf and F. L. Lewis, "Nearly optimal control laws for nonlinear systems with saturating actuators using a neural network HJB approach," *Automatica*, vol. 41, no. 5, pp. 779–791, 2005.

[46] B. A. Finlayson, *The Method of Weighted Residuals and Variational Principles*. New York: Academic, 1972.

[47] B. Stevens and F. L. Lewis, *Aircraft Control and Simulation*, 2nd ed. New York: Wiley, 2003.

[48] K. G. Vamvoudakis, "Online learning algorithms for differential dynamic games and optimal control," Ph.D. dissertation, Faculty of Graduate School, Univ. Texas at Arlington, Arlington, 2011.

[49] V. Nevisti'C and J. A. Primbs, "Constrained nonlinear optimal control: A converse HJB approach," Dept. Control & Dynamical Syst., California Inst. Technology, Pasadena, Tech. Rep. TR96-021, 1996.

Huai-Ning Wu was born in Anhui, China, on November 15, 1972. He received the B.E. degree in automation from the Shandong Institute of Building Materials Industry, Jinan, China, and the Ph.D. degree in control theory and control engineering from Xi'an Jiaotong University, Xi'an, China, in 1992 and 1997, respectively.

He was a Post-Doctoral Researcher with the Department of Electronic Engineering, Beijing Institute of Technology, Beijing, China, from 1997 to 1999. He joined the School of Automation Science and Electrical Engineering, Beihang University (formerly Beijing University of Aeronautics and Astronautics), Beijing, in 1999. From 2005 to 2006, he was a Senior Research Associate with the Department of Manufacturing Engineering and Engineering Management (MEEM), City University of Hong Kong, Kowloon, Hong Kong, where he was a Research Fellow from 2006 to 2008 and from July to August 2010. From July to August in 2011, he was a Research Fellow with the Department of Systems Engineering and Engineering Management, City University of Hong Kong. He is currently a Professor with Beihang University. His current research interests include robust control and filtering, fault-tolerant control, distributed parameter systems, and fuzzy and neural modeling and control.

Dr. Wu is a member of the Committee of Technical Process Failure Diagnosis and Safety of the Chinese Association of Automation.

Biao Luo received the B.E. degree in measuring and control technology and instrumentations and the M.E. degree in control theory and control engineering from Xiangtan University, Xiangtan, China, in 2006 and 2009, respectively. He is currently pursuing the Ph.D. degree in control science and engineering with Beihang University (formerly Beijing University of Aeronautics and Astronautics), Beijing, China.

His current research interests include distributed parameter systems, optimal control, data-based control, fuzzy and neural modeling and control, hypersonic entry and re-entry guidance, reinforcement learning, approximate dynamic programming, and evolutionary computation.

Mr. Luo was a recipient of the Excellent Master Dissertation Award of Hunan Province in 2011.