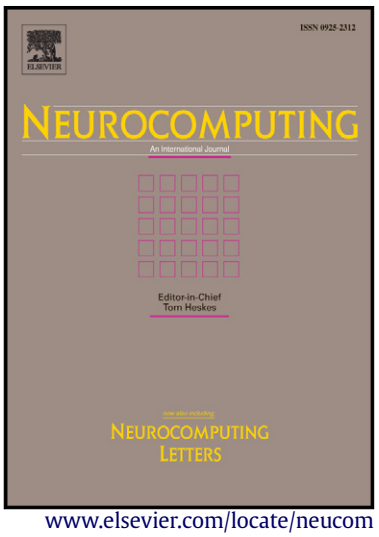


www.elsevier.com/locate/neucom

Dual Heuristic Dynamic Programming for Nonlinear Discrete-Time Uncertain Systems with State Delay

Bin Wang, Dongbin Zhao, Cesare Alippi, Derong Liu

PII: S0925-2312(13)00739-X
DOI: http://dx.doi.org/10.1016/j.neucom.2013.06.037
Reference: NEUCOM13534

To appear in: *Neurocomputing*

Received date: 9 March 2013
Revised date: 23 May 2013
Accepted date: 2 June 2013

Cite this article as: Bin Wang, Dongbin Zhao, Cesare Alippi, Derong Liu, Dual Heuristic Dynamic Programming for Nonlinear Discrete-Time Uncertain Systems with State Delay, *Neurocomputing*, http://dx.doi.org/10.1016/j.neucom.2013.06.037

Dual Heuristic Dynamic Programming for Nonlinear Discrete-Time Uncertain Systems with State Delay☆

Bin Wang[a], Dongbin Zhao[a], Cesare Alippi[a,b], Derong Liu[a]

[a]*The State Key Laboratory of Management and Control for Complex Systems, Institute of Automation, Chinese Academy of Sciences, Beijing 100190, China*
[b]*Dipartimento di Elettronica e Informazione, Politecnico, di Milano, 20133 Milano, Italy*

Abstract

The paper proposes a novel iterative control scheme based on neural networks for optimally controlling a large class of nonlinear discrete-time systems affected by an unknown time variant delay and system uncertainties. An iterative Dual Heuristic dynamic Programming (DHP) algorithm has been envisaged to design the controller which is proven to converge to the optimal one. The key elements required by the DHP, namely the performance index function, the optimal control policy and the nonlinear delay-affected discrete-time system are modeled with feedforward neural networks. Examples demonstrate the validity of the proposed optimal control approach and its effectiveness in dealing with nonlinear time delay situations.

Keywords: Dual Heuristic dynamic Programming (DHP), time delay, nonlinear uncertain system, discrete-time, neural networks

1. Introduction

Time delay is a widespread phenomenon in industrial processes either arising from the inherent time delay introduced by the elements composing the system, or from intentional actions considered for control purposes [1],[2].Because of time delay, state variables cannot timely reflect changes in the system, resulting in reduced performance of the controller. The presence of time delay in a process makes the analysis and design of the control system a complex task which, however, cannot be avoided. In this direction, [3] and [4] proposed a PID control approach to control time delay systems. In [5] an algebraic Riccati equation approach is presented to derive the memoryless linear state feedback control scheme for uncertain dynamic delay systems. [6] developed a stabilizing controller for a class of time delay nonlinear systems based on the constructive use of appropriate Lyapunov-Krasovskii functionals. Linear matrix inequality (LMI) method is used in [7, 8] to design state-feedback controller for the linear time delay systems and a new Lyapunov-Krasovskii functional is proposed for robust stability analysis. [9] used the T-S fuzzy model to represent the state-space model of nonlinear discrete-time systems with time delays and a stable fuzzy $H\infty$ filter was designed for signal estimation. In [10] the discrete time delay system is transformed into a non-delayed system by a function-based transformation; an optimal tracking controller is constructed by solving Riccati matrix equation and Stein matrix equations. A simultaneous state and disturbance estimation technique is developed for time delay systems and applied to fault estimation and signal compensation in [11]. However, the above theories and methods are either limited to linear time delay systems or transform the nonlinear time delay systems into linear ones by fuzzy method or robust method, which may cause oscillation in the case of large or a time-variant delay. Generally the optimal control of time delay systems is an infinite-dimensional control problem [12], which is very difficult to deal with, thus some advanced control

☆This work was supported partly by National Natural Science Foundation of China under Grant Nos. 61273136, and 61034002, Beijing Natural Science Foundation under Grant No. 4122083, and Visiting Professorship of Chinese Academy of Sciences.

*Corresponding author at: The State Key Laboratory of Management and Control for Complex Systems, Institute of Automation, Chinese Academy of Sciences, Beijing, 100190, PR China. Tel.: +8613683277856, fax:8610-8261-9580.

Email addresses: bin.wang@ia.ac.cn (Bin Wang), dongbin.zhao@ia.ac.cn (Dongbin Zhao), alippi@elet.polimi.it (Cesare Alippi), derong.liu@ia.ac.cn (Derong Liu)

methods must be developed to design the optimal controller.

In [13] an approach based on Adaptive Dynamic Programming (ADP) has been proposed which is shown to be effective in solving dynamic programming problems by learning a model approximating the Bellman equation [14]. More in detail, ADP builds the '*Critic*' and the '*Action*' functions, to approximate the value and the control policy functions, respectively. There are several main variants of ADP [15, 16]: Heuristic Dynamic Programming (HDP), Dual Heuristic dynamic Programming (DHP), Globalized Dual Heuristic dynamic Programming (GDHP), which differentiate in the output provided by the critic function. A first attempt to use ADP to deal with the optimal control of time delay systems can be found in [12] and [17] where an optimal control scheme for a class of nonlinear systems characterized by a time delay in the state and control variables was developed. [18] proposed a new iterative HDP algorithm to solve the optimal control problem for a class of nonlinear discrete time-delay systems with saturating actuators. An HDP algorithm was also proposed in [19] to solve the optimal tracking control problem for a class of nonlinear discrete-time systems with time delays. For the optimal control problem of a class of nonlinear time-delay systems with control constraints, [20] introduced a nonquadratic performance functional to overcome the control constraints and proposed an iterative HDP algorithm to design the optimal feedback controller.

Different from HDP, where a critic network is built to model the cost function, in DHP the partial derivatives of the cost function with respect to its inputs are modeled instead [21], [22]. The critic network of DHP hence provides a vectored output in-stead of a scalar as given by the HDP, and each component of the vector is used to regulate the parameters of the action separately. It comes out that DHP is also more appropriate to characterize a multiple input-output system. Up to the best of our knowledge, there are no methods proposing a solution of the optimal control problem for time delay systems by means of a DHP technique. This motivates our research.

More specifically, in this paper we propose a novel iterative DHP algorithm to optimally control a relevant class of nonlinear discrete-time systems affected by time delay and system uncertainties. Moreover, it is proved that the iterative algorithm converges to the optimal controller for the class of systems. The iterative DHP algorithm models the requested functions through feedforward neural networks, as suggested in [23].

The paper is organized as follows. Section 2 formalizes the control problem. The optimal control scheme based on iterative DHP algorithm is developed in section 3. Section 4 discusses the neural network implementation for the iterative DHP algorithm. Finally, experimental results are given in section 5.

2. Problem Formulation

Consider the relevant class of nonlinear discrete-time delay systems proposed in [8]

$$x(k+1) = (A+\Delta A)x(k) + (A_d+\Delta A_d)x(k-\tau(k)) + (B+\Delta B)u(k) + f(x(k), x(k-\tau(k))) \quad (1)$$

where $x(k) \in R^n$ and $u(k) \in R^m$ denote the state and input vectors at time instant k, respectively. A, A_d and B are constant matrices of appropriate dimensions. ΔA, ΔA_d and ΔB are uncertain matrices. $f(x(k), x(k-\tau(k))) : R^n \times R^n \rightarrow R^n$ is the nonlinear mapping also function of the delayed states. $\tau(k)$ is the unknown time-varying delay function bounded by constant b, such that $0 \leq \tau(k) \leq b$. The initial state is given by $x(s) = \phi(s)$, $-b \leq s \leq 0$; we assume system (1) to be controllable [24].

Remark 1 (taken from [8]). The uncertain parameters are assumed to be bounded with the form $[\Delta A \ \Delta A_d \ \Delta B] = HF[E_1 \ E_2 \ E_d]$, where F is an unknown matrix satisfying $F^TF \leq I$, E_1, E_2, E_d and H are constant matrices describing the uncertainty structure. The nonlinear uncertainty f is assumed to satisfy

$$f^T f \leq \begin{bmatrix} x^T(k) \\ x^T(k-\tau(k)) \end{bmatrix}^T \begin{bmatrix} H_1^T H_1 & 0 \\ 0 & H_2^T H_2 \end{bmatrix} \begin{bmatrix} x^T(k) \\ x^T(k-\tau(k)) \end{bmatrix}$$

Remark 2. It should be noted that many physical processes are governed by nonlinear differential equations of the form (1), e.g., the cold rolling mills [2] and recycled chemical reactors [25]. (1) refers to time delays affecting the system state only, which is one of the common circumstances arising in many real systems. Consider the quadratic performance index function:

$$V(x(k)) = \sum_{k=0}^{\infty} \{x^T(k)Qx(k) + u^T(k)Ru(k) - \gamma^2 x^T(k-\tau(k))x(k-\tau(k))\} \quad (2)$$

where Q and R are positive definite matrices and γ is a prescribed positive constant. Optimal control requires the identification of the control policy $u(x(k)) = u(k)$ that minimizes equation (2). Let $V^*(x(k))$ be the optimal performance index function

$$V^*(x(k)) = \min_{u(k)} V(x(k)) \quad (3)$$

We comment that the state feedback control policy $u(k)$ must stabilize the system (1) on $\Re^n$, as well as guarantee that the performance index function (2) is finite. From [26, 27] a control policy $u(k)$ is defined to be admissible with respect to function

$$V(x(k)) = \sum_{i=k}^{\infty} \{x^T(i)Qx(i) + u^T(i)Ru(i)\}$$

if $u(k)$ is continuous and stabilizes (1) on $\Re^n$, $u(0) = 0$, and $\forall x(0) \in \Re^n$, $V(x(0))$ is finite. Since (2) is a special case of the above function and hypotheses are satisfied, the control policy for (2) is admissible. Equation (2) can be rewritten as

$$\begin{aligned} V(x(k)) &= x^T(k)Qx(k) + u^T(k)Ru(k) \\ &\quad - \gamma^2 x^T(k-\tau(k))x(k-\tau(k)) + V(x(k+1)) \end{aligned} \quad (4)$$

According to Bellman's optimality principle [28], it follows that the optimal performance index function $V^*(x(k))$ satisfies the discrete-time Hamilton-Jacobi-Bellman (DTHJB) equation

$$\begin{aligned} V^*(x(k)) &= \min_{u(k)}\{x^T(k)Qx(k) + u^T(k)Ru(k) \\ &\quad - \gamma^2 x^T(x-\tau(k))x(k-\tau(k)) + V^*(x(k+1))\} \end{aligned} \quad (5)$$

and the corresponding optimal control policy $u^*(k)$ is

$$\begin{aligned} u^*(x(k)) &= arg\min_{u(k)}\{x^T(k)Qx(k) + u^T(k)Ru(k) \\ &\quad - \gamma^2 x^T(x-\tau(k))x(k-\tau(k)) + V^*(x(k+1))\} \end{aligned} \quad (6)$$

The optimal control law $u^*(k)$ follows by differentiating the argument of the $\min_{u(k)}$ function of (5) with respect to $u(k)$, i.e.,

$$\begin{aligned} &\frac{\partial(x^T(k)Qx(k) + u^T(k)Ru(k))}{\partial u(k)} \\ &-\frac{\gamma^2 x^T(x-\tau(k))x(k-\tau(k)))}{\partial u(k)} \\ &+(\frac{\partial x(k+1)}{\partial u(k)})^T \frac{\partial V^*(x(k+1))}{\partial x(k+1)} = 0 \end{aligned}$$

from which we obtain

$$u^*(k) = -\frac{1}{2}R^{-1}(B+\Delta B)^T \frac{\partial V^*(x(k+1))}{\partial x(k+1)} \quad (7)$$

3. Optimal Control Based on an Iterative DHP Algorithm

3.1. Derivation of the iterative DHP algorithm

The iterative DHP algorithm can be derived by relying on the greedy iteration principle where we update both the value function and control policy at the same iteration. Start with the initial value $V_0(\cdot) = 0$. The control vector $u_0(k)$ can be computed as

$$\begin{aligned} u_0(x(k)) &= arg\min_{u(k)}\{x^T(k)Qx(k) + u^T(k)Ru(k) \\ &\quad - \gamma^2 x^T(x-\tau(k))x(k-\tau(k)) + V_0(x(k+1))\} \end{aligned} \quad (8)$$

By knowing the control policy $u_0(k)$ we can compute the performance index function $V_1(x(k))$

$$\begin{aligned} V_1(x(k)) &= \min_{u(k)}\{x^T(k)Qx(k) + u^T(k)Ru(k) \\ &\quad - \gamma^2 x^T(x-\tau(k))x(k-\tau(k)) + V_1(x(k+1))\} \\ &= x^T(k)Qx(k) + u_0^T(k)Ru_0(k) \\ &\quad - \gamma^2 x^T(x-\tau(k))x(k-\tau(k)) \end{aligned} \quad (9)$$

The state vector is then updated as

$$\begin{aligned} x_0(k+1) &= (A+\Delta A)x_0(k) + (A_d+\Delta A_d)x_0(k-\tau(k)) \\ &\quad + (B+\Delta B)u_0(k) + f(x_0(k), x_0(k-\tau(k))) \end{aligned} \quad (10)$$

The algorithm then iterates over index i yielding the control policy

$$\begin{aligned} u_i(x(k)) &= arg\min_{u(k)}\{x^T(k)Qx(k) + u^T(k)Ru(k) \\ &\quad - \gamma^2 x^T(x-\tau(k))x(k-\tau(k)) + V_i(x(k+1))\} \\ &= -\frac{1}{2}R^{-1}(B+\Delta B)^T \frac{\partial V_i(x(k+1))}{\partial x(k+1)} \end{aligned} \quad (11)$$

associated with the performance index function

$$\begin{aligned} V_{i+1}(x(k)) &= \min_{u(k)}\{x^T(k)Qx(k) + u^T(k)Ru(k) \\ &\quad - \gamma^2 x^T(x-\tau(k))x(k-\tau(k)) + V_i(x(k+1))\} \\ &= x^T(k)Qx(k) + u_i^T(k)Ru_i(k) \\ &\quad - \gamma^2 x^T(x-\tau(k))x(k-\tau(k)) + V_i(x(k+1)) \end{aligned} \quad (12)$$

and the state vector

$$\begin{aligned} x_i(k+1) &= (A+\Delta A)x_i(k) + (A_d+\Delta A_d)x_i(k-\tau(k)) \\ &\quad + (B+\Delta B)u_i(k) + f(x_i(k), x_i(k-\tau(k))) \end{aligned} \quad (13)$$

We recall that k is the time index and i the iteration index for the control policy and the performance index function. We now need to prove that the suggested iterative algorithm converges to the optimal control solution. This will be done in the next section.

3.2. Convergence proof of the iterative DHP algorithm

To demonstrate the convergence of the algorithm proposed in section 3.1, we follow the framework delineated in [26, 27]. At first we show that value V_i in (11) converges to V^* and that u_i in (12) converges to u^* as i tends to infinity.

Lemma1

Let V_i be the performance index function of (12) and Λ_i defined with the recurrent form

$$\begin{aligned}\Lambda_{i+1}(x(k)) = {} & x^T(k)Qx(k) + \mu_i^T(k)R\mu_i(k) \\ & - \gamma^2 x^T(k-\tau(k))x(k-\tau(k)) + \Lambda_i(x(k+1))\end{aligned} \tag{14}$$

For any arbitrary sequence of control policies $\{\mu_i\}$ and policies $\{u_i\}$ in (11), if $V_0(\cdot) = \Lambda_0(\cdot) = 0$, then $V_{i+1}(x(k)) \le \Lambda_{i+1}(x(k)), \forall i$.

Proof. The proof immediately follows by noting that V_{i+1} can be obtained by minimizing the right hand formula of (12) with respect to u_i, while Λ_{i+1} is a result of any arbitrary control input.

Lemma2

Given sequence $\{V_{i+1}\}$ of (12), if the system is controllable, then there it exists an upper bound ε so that $0 \le V_{i+1}(x(k)) \le \varepsilon, \forall i$.

Proof. Choose $\{\bar{\mu}_i\}$ as any stabilizing and admissible control policy sequence, V_i as in (12) and $\bar{\Lambda}_i$

$$\begin{aligned}\bar{\Lambda}_{i+1}(x(k)) = {} & x^T(k)Qx(k) + \bar{\mu}_i^T(k)R\bar{\mu}_i(k) \\ & - \gamma^2 x^T(k-\tau(k))x(k-\tau(k)) + \bar{\Lambda}_i(x(k+1))\end{aligned} \tag{15}$$

with $V_0(\cdot) = \bar{V}_0(\cdot) = 0$, then, we have that

$$\begin{aligned}\bar{\Lambda}_{i+1}(x(k)) - \bar{\Lambda}_i(x(k)) &= \bar{\Lambda}_i(x(k+1)) - \bar{\Lambda}_{i-1}(x(k+1)) \\ &= \bar{\Lambda}_{i-1}(x(k+2)) - \bar{\Lambda}_{i-2}(x(k+2)) \\ &\vdots \\ &= \bar{\Lambda}_1(x(k+i)) - \bar{\Lambda}_0(x(k+i))\end{aligned} \tag{16}$$

Since $\bar{\Lambda}_0(\cdot) = 0$, (16) can be rewritten as

$$\begin{aligned}\bar{\Lambda}_{i+1}(x(k)) &= \bar{\Lambda}_1(x(k+i)) + \bar{\Lambda}_i(x(k)) \\ &= \bar{\Lambda}_1(x(k+i)) + \bar{\Lambda}_1(x(k+i-1)) + \bar{\Lambda}_{i-1}(x(k)) \\ &\quad \vdots \\ &= \bar{\Lambda}_1(x(k+i)) + \bar{\Lambda}_1(x(k+i-1)) \\ &\quad + \bar{\Lambda}_1(x(k+i-2)) + \cdots + \bar{\Lambda}_1(x(k))\end{aligned} \tag{17}$$

(17) becomes

$$\begin{aligned}\bar{\Lambda}_{i+1}(x(k)) &= \sum_{j=0}^{i} \bar{\Lambda}_1(x(k+j)) \\ &\le \sum_{j=0}^{\infty} \{x^T(k+j)Qx(k+j) + \bar{\mu}^T(k+j)R\bar{\mu}(k+j) \\ &\quad - \gamma^2 x^T(k+j-\tau(k))x(k+j-\tau(k))\}\end{aligned} \tag{18}$$

Since $\bar{\mu}_i$ is a stabilizing and admissible control input ($x(k) \to 0\, as\, k \to \infty$),

$$\bar{\Lambda}_{i+1}(x(k)) \le \sum_{j=0}^{\infty} \bar{\Lambda}_1(x(k+j)) \le \varepsilon, \forall i. \tag{19}$$

holds. By applying Lemma 1, we have that

$$V_{i+1}(x(k)) \le \bar{\Lambda}_{i+1}(x(k)) \le \varepsilon, \forall i. \tag{20}$$

We now present the main theorem.

Theorem1

Define the sequence $\{V_i\}$ as in (12) with $V_0(\cdot) = 0$. Then $\{V_i\}$ is a nondecreasing sequence satisfying inequality $V_{i+1}(x(k)) \ge V_i(x(k)), \forall i$, and converging to the optimal value function of the DTHJB equation (5), i.e., $V_i \to V^*\, as\, i \to \infty$.

Proof. Define $\tilde{\Lambda}_i$ as

$$\begin{aligned}\tilde{\Lambda}_{i+1}(x(k)) = {} & x^T(k)Qx(k) + u_{i+1}^T(k)Ru_{i+1}(k) \\ & - \gamma^2 x^T(k-\tau(k))x(k-\tau(k)) + \tilde{\Lambda}_i(x(k+1))\end{aligned} \tag{21}$$

where $V_0(\cdot) = \tilde{\Lambda}_0(\cdot) = 0$, and the control policy u_i as in (11). The proof follows in two steps. At first we show that $\tilde{\Lambda}_i(x(k)) \le V_{i+1}(x(k))$ by induction.

Since

$$V_1(x(k)) - \tilde{\Lambda}_0(x(k)) = x^T(k)Qx(k) \ge 0$$

we have that

$$V_1(x(k)) \ge \tilde{\Lambda}_0(x(k))$$

Assume now that $V_i(x(k)) \ge \tilde{\Lambda}_{i-1}(x(k)), \forall x(k)$. Then from (12) and (21), we obtain

$$V_{i+1}(x(k)) - \tilde{\Lambda}_i(x(k)) = V_i(x(k+1)) - \tilde{\Lambda}_{i-1}(x(k+1)) \ge 0$$

from which

$$V_{i+1}(x(k)) \ge \tilde{\Lambda}_i(x(k)) \tag{22}$$

Lemma 1 now grants that $V_{i+1}(x(k)) \le \tilde{\Lambda}_i(x(k))$ and, therefore

$$V_i(x(k)) \le \tilde{\Lambda}_i(x(k)) \le V_{i+1}(x(k))$$

namely,

$$V_i(x(k)) \leq V_{i+1}(x(k))$$

$\{V_i\}$ is a nondecreasing sequence bounded thanks to Lemma 2.As such, we can conclude that $V_i \rightarrow V^* \, as \, i \rightarrow \infty$.

We just proved that the value function sequence of the DTHJB equation converges to the optimal value. Correspondingly, the control sequence also converges to the optimal one.

3.3. The proposed iterative DHP algorithm

1. Give the initial states $x(s) = \phi(s), -m \leq s \leq 0$, the maximum number of iterations i_{max} and the computation accuracy ε;
2. Set $i = 0$; $V_0(\cdot) = 0$;
3. Compute $u_0(k)$ according to (8); compute the performance index function $V_1(x(k))$ as in (9);
4. Compute the next state $x_0(k+1)$ according to (10);
5. do {
6. $i = i + 1$; Compute $u_i(k)$ for $i \geq 1$ as in (11); Update the state vector $x_i(k+1)$ by (13);
7. Compute the value $V_i(x(k))$ according to (12);
8. }while ($\|V_{i+1}(x(k)) - V_i(x(k))\| \geq \varepsilon$ and $i \leq i_{max}$)

Algorithm 1: the iterative DHP algorithm

Starting from the initial state vector $X = [x_0(k) \quad x_0(k - \tau(k))]^T$ the DHP algorithm iterates. With $V_0(\cdot) = 0$, we can get $u_0(k)$ by (8). Then the next value function $V_1(x(k))$ can be computed as in (9). The next state vector can be obtained by (10). Step by step, we can compute $u_i(k)$ as in (11), update the state vector $x_i(k+1)$ by (13), and compute the value $V_i(x(k))$ according to (12). In this way the algorithm iterates until the value function $V(x(k))$ converges to a small constant ε. To make sure the iteration procedure is continuous, we experimentally choose a maximum number of iterations i_{max}. If the value function can't converge to ε, which means that the trail failed, i_{max} can be used to avoid the stuck of the training process.

4. Neural Networks Implementation for the Iterative DHP Algorithm

As with [23] we approximate the control policy $u_i(k)$ and the performance index function $V_i(x(k))$ with static feedforward neural networks.

Figure 1 presents a schematic diagram of the iterative DHP algorithm. There are three components, each of

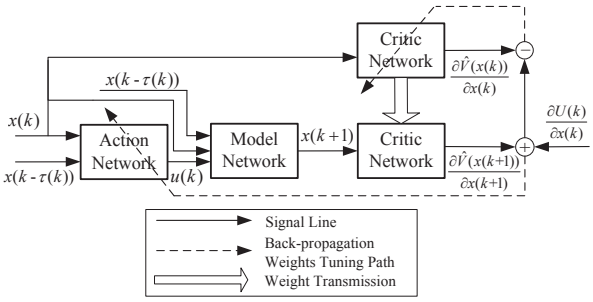


Figure 1: A schematic representation of the DHP algorithm.

which modeled with a neural network: the model network, the critic network, and the action network, which approximate the nonlinear discrete-time delay system, the performance index function and the optimal control policy, respectively. In this paper, all the three networks are chosen as feedforward neural networks with a single hidden layer and a linear output. The approximating functions exposed by the network is

$$\hat{F}(X, v, w) = w^T \sigma(v^T X) \tag{23}$$

where X is the input vector with appropriate dimension, v is the weight matrix between the input layer and the hidden layer and w is the weight matrix between the hidden layer and the output layer [23]. The activation function of the hidden units $\sigma(v^T X) \in \mathbb{R}^l$ is chosen to be the hyperbolic tangent. The number of hidden units l is determined with a trial and error approach so as to minimize the validation set.

The training procedure provides the weights v^* and w^* and, hence, the approximation function $F(X, v^*, w^*)$. Networks are trained with a gradient-based algorithm.

4.1. The Model network

The model network is used to model the system to be controlled, As such, the model network must be configured before carrying out the iterative algorithm. The inputs to the model network are $x(k)$, $x(k - \tau(k))$ and $u(k)$, and produce output

$$\hat{x}_i(k+1) = \hat{F}_i(X, v, w) \tag{24}$$

where $X = [x_i(k) \quad x_i(k - \tau(k)) \quad u_i(k))]^T$.

The weights of the model network are frozen once training is perfected.

4.2. The Critic network

The critic network approximates the partial derivatives of the performance index function, $\lambda_i(x(k)) =$

$\partial V_i(x(k))/\partial x(k)$. The input is the state variable $x(k)$, the output of the critic network is

$$\hat{\lambda}_i(k) = \hat{F}_i(x(k), v, w) \tag{25}$$

Once the critic and the model networks have been configured, the performance index function can be computed as

$$\begin{aligned}\lambda_{i+1}(x(k)) = & \frac{\partial(x^T(k)Qx(k) + u_i^T(k)Ru_i(k)}{\partial x(k)} \\ & - \frac{\partial(\gamma^2 x^T(x-\tau(k))x(k-\tau(k)))}{\partial x(k)} + \frac{\partial \hat{V}_i(x(k+1))}{\partial x(k)} \\ & = 2Qx(k) + (\frac{\partial x(k+1)}{\partial x(k)})^T \hat{\lambda}_i(x(k+1)) \end{aligned} \tag{26}$$

4.3. The Action network

The action network is needed to approximate the control policy $u_i(k)$. The states $x(k)$, $x(k-\tau(k))$ are used as inputs vector to obtain the optimal control $\hat{u}_i(k)$ and the output is

$$\hat{u}_i(k) = \hat{F}_i(X, v, w) \tag{27}$$

where $X = [x_i(k) \quad x_i(k-\tau(k))]^T$. Training is computed by considering the reference control input

$$u_i(k) = -\frac{1}{2}R^{-1}(B + \Delta B)^T \hat{\lambda}_i(x(k+1)) \tag{28}$$

5. Experiments

An experimental campaign is carried out to demonstrate the effectiveness of the iterative DHP algorithm in solving the optimal control problems of the delayed nonlinear system. In the following we first choose a typical example and a three order system to verify the effectiveness of the proposed DHP algorithm, and then apply the approach to the typical two-stage chemical reactor with delayed recycle streams.

5.1. Example 1: A typical case

Consider the nonlinear discrete-time delay system (1):

$$A = \begin{bmatrix} 0 & 1 \\ -1 & 1 \end{bmatrix}, B = \begin{bmatrix} 0 \\ 1 \end{bmatrix},$$

$$A_d = \begin{bmatrix} 0.3 & 0.2 \\ 0.2 & 0.3 \end{bmatrix}, H = \begin{bmatrix} 0.1 & 0 & 0.1 \\ 0 & 1 & 0 \end{bmatrix},$$

$$E_1 = \begin{bmatrix} 1 & 1 \\ 0 & 0 \\ 0 & 0 \end{bmatrix}, E_2 = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}, E_d = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 1 & 1 \end{bmatrix}, F = cos(k),$$

The time delay is set to be $\tau(k) = 2$ for training. The initial state is $x(k) = [1 \quad -1]^T$ for $-2 \le k \le 0$. The performance index function is defined as (2), where $Q = R = I$, $\gamma^2 = 0.1$. In order to implement the iterative DHP algorithm, we adopt similar experimental setup as in [17], similarly hereinafter. We selected three-layer feedforward neural networks to approximate the model network, the critic network, and the action network. The structures of the networks are 5-8-2,2-8-2,4-8-1, respectively. Model network training was performed over 10000 data. Similarly, we trained the critic and the action networks. The initial weights of the neural networks are all chosen randomly in $[-0.5, 0.5]$.

Table 1: The performance comparison of DHP and HDP

Delay times	DHP		HDP	
$\tau(k)$	t_s(time steps)	e_{ss}(%)	t_s(time steps)	e_{ss}(%)
1	20	0	31	0
2	24	0	40	0
Rand[1,5]	63	0.37	$--$[1]	$--$

1. $'--'$ means HDP cannot stabilize in 100 time steps. Similarly hereinafter.

To compare results we implemented the iterative HDP algorithm of [17]. After training, we applied the optimal control law designed by the iterative DHP algorithm and the corresponding HDP one for 100 time steps. The main performance indexes such as the settling time t_s and the steady-state error e_{ss} are considered, and the delay increases from 1 to 4 time steps. To further verify the validity and the robustness of the designed optimal control law, we set the delay $\tau(k)$ as a time-variant discrete function, which outputs a random integral following an uniform distribution within the [1,5] interval. The performances of the iterative DHP and HDP algorithms are given in Table 1. The state trajectories are shown in Figure 2 and Figure 4, respectively.

It emerges that the suggested iterative DHP algorithm performs better than the HDP one. By varying the delay, the DHP always needs less settling time than the HDP one, and the DHP has little steady-state error. Furthermore, the time-variant delay has been dealt with effectively, which confirms the robustness of the iterative DHP algorithm.

Remark 3. It should be noted that when time-variant delays (see Figure 3) are added into the system, the DHP algorithm cannot succeed in each simulation. In fact, severe oscillation can occur due to the uncertain time delay. Nevertheless, the DHP always performs better than HDP.

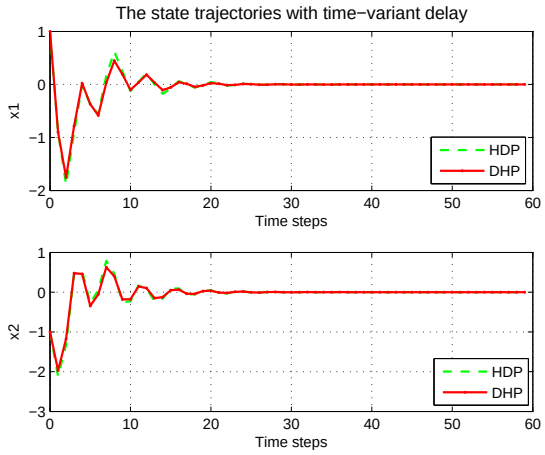


Figure 2: The state trajectories x with constant time delay ($\tau = 2$).

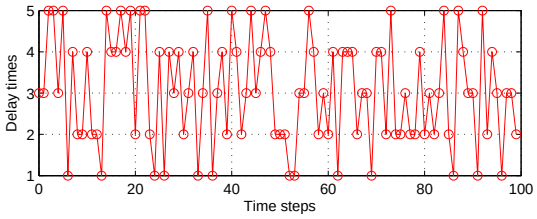


Figure 3: The time-variant delay.

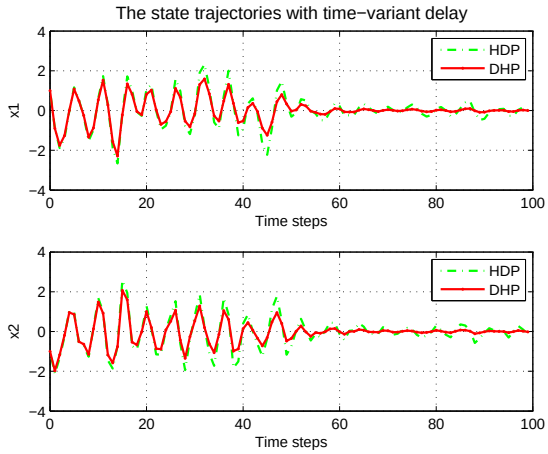


Figure 4: The state trajectories x with time-variant delay.

5.2. *Example 2: A three order system*

Consider the three order nonlinear discrete-time delay system (1)(modified from [8]):

$$A = \begin{bmatrix} 0.7 & 0 & -0.5 \\ 0.05 & 0.8 & 0 \\ 0 & 0.3 & 0.6 \end{bmatrix}, B = \begin{bmatrix} 0.3 \\ 0 \\ 0.6 \end{bmatrix},$$

$$A_d = \begin{bmatrix} -0.2 & 0 & 0 \\ 0 & -0.1 & 0.1 \\ 0 & 0 & -0.2 \end{bmatrix}, H = \begin{bmatrix} 0.1 & 0 & 0.2 \end{bmatrix},$$

$$E_1 = \begin{bmatrix} 0.2 & 0 & 0.3 \end{bmatrix}, E_2 = 0.4, E_d = 0, F = sin(k),$$

$$f(x(k), x(k-\tau(k))) = \begin{bmatrix} x_2(k-\tau(k)) \cdot x_3(k-\tau(k)) \\ 0 \\ x_1(k-\tau(k)) \cdot sin(x_2(k-\tau(k))) \end{bmatrix}.$$

The time delay is set to be $\tau(k) = 2$ during training. The initial state is $x(k) = [1 \quad -1 \quad 1]^T$ for $-2 \le k \le 0$. The performance index function is defined as (2), where $Q = R = I$, $\gamma^2 = 0.1$. We trained the relevant neural networks as above.

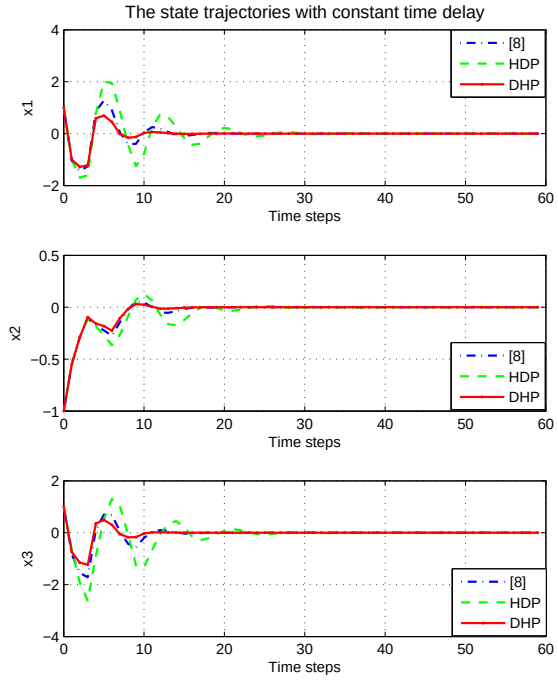


Figure 5: The state trajectories x with constant time delay($\tau = 2$).

To compare results we implemented the iterative HDP algorithm of [17] and the optimal guaranteed cost controller of [8]. The performances of the iterative DHP, the HDP and the optimal guaranteed cost control algorithms are given in Table 2. The state trajectories are shown in Figure 5 and Figure 6, respectively.

Table 2 shows that with different constant time delay, the DHP always need less settling time and has little steady-state error compared with the HDP one and the guaranteed cost controller. When dealing with time-variant delay (see Rand[4, 8] in Table 2), the DHP still

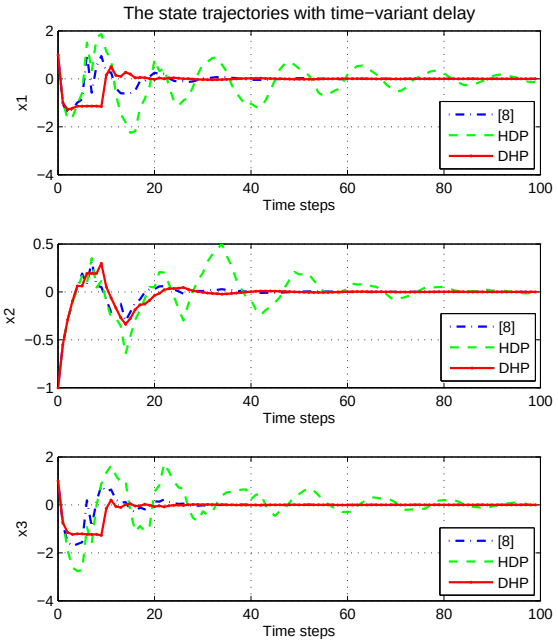


Figure 6: The state trajectories x with time-variant delay.

Table 2: The performance comparison of DHP, HDP and guarnteed cost control

Delay times	DHP		HDP		guaranteed cost	
$\tau(k)$	t_s	$e_{ss}(\%)$	t_s	$e_{ss}(\%)$	t_s	$e_{ss}(\%)$
1	12	0	13	0	15	0
2	14	0	21	0	21	0
3	18	0	25	0	26	0
4	21	0	41	0	35	0
Rand[4,8]	57	0.56	−−	−−	61	0.67

performs fairly well as the guaranteed cost controller. Once again, the proposed controller is able to deal with the time-variant delay, hence showing the robustness of the iterative DHP algorithm.

5.3. *Example 3: A two-stage chemical reactor with delayed recycle streams*

A practical example of a two-stage chemical reactor with delayed recycle streams is considered as a third example. The mass balance equations governing the reactor shown in Figure 7 [11] are:

$$\begin{cases} \dot{x}_1(t) &= -\frac{1}{\theta_1}x_1(t) - a_1 x_1(t) + \frac{1-R_2}{V_1}x_2(t) + \delta_1(t, x_2(t-\tau)) \\ \dot{x}_2(t) &= -\frac{1}{\theta_2}x_2(t) - a_1 x_1(t) + \frac{R_1}{V_2}x_1(t-\tau) + \frac{R_2}{V_2}x_2(t-\tau) \\ & \quad + \frac{G}{V_2}u + \delta_2(t, x_1(t-\tau)) \end{cases}$$

where x_1 and x_2 are the reaction compositions, θ_1 and θ_2 are the reactor residence times, a_1 and a_2 are the reaction constants, R_1 and R_2 are the recycle flow rate, V_1 and V_2 are the reactor volumes, G is the feed rate, δ_1 and δ_2 are uncertain nonlinear functions with time delay. The parameters are given as follows: $\theta_1 = \theta_2 = 1$, $a_1 = a_2 = 1$, $R_1 = R_2 = 0.5$, $V_1 = V_2 = 1$, $\tau = 2$, $\delta_1(t, x_2(t-\tau)) = \rho sin(t)x_2^2(t-\tau)$, $\delta_2(t, x_1(t-\tau)) = \rho sin(t)x_1^2(t-\tau)$. We define ρ as amplitude of the uncertainties, which means that different ρ has different effect to the system.

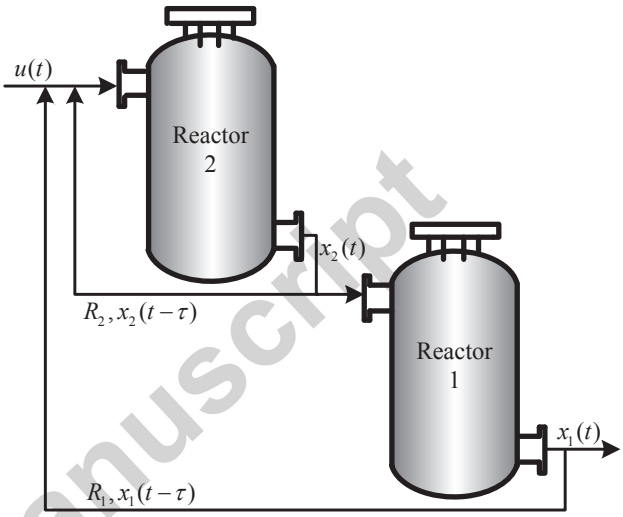


Figure 7: Two-stage chemical reactor with delayed recycle streams.

Discretization of the above system is firstly made. By applying the iterative DHP algorithm proposed in the paper, the results are shown in Figure 8 and Figure 9. The state trajectories show that the system with constant time delay can be controlled to stable in time. For the time variant delay, the controller performs well with acceptable oscillation.

Table 3: The performance comparison of DHP and reference [11] with different uncertainties

Uncertainties	DHP		Feedback control in [11]	
ρ	t_s	$e_{ss}(\%)$	t_s	$e_{ss}(\%)$
0.1	23	0	47	0
0.2	31	0.02	53	0.08
0.3	93	0.13	NaN	NaN
0.4	118	0.30	NaN	NaN
0.5	NaN	NaN	NaN	NaN

To further compare the robust performance of the iterative approach we consider the output feedback controller presented in [11]. The control input is $u = -Fx_1+v$, where v is the reference input, and F is the output feedback gain. Here, we set $v = 0$, $F = 4$. Results are given in Table 3. When the uncertainties increase,

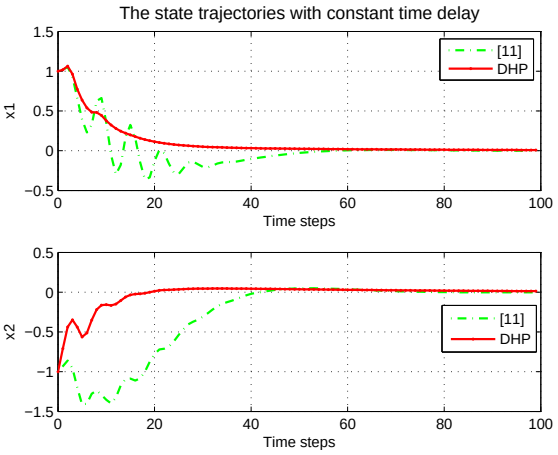


Figure 8: The state trajectories x with constant time delay ($\tau = 2$).

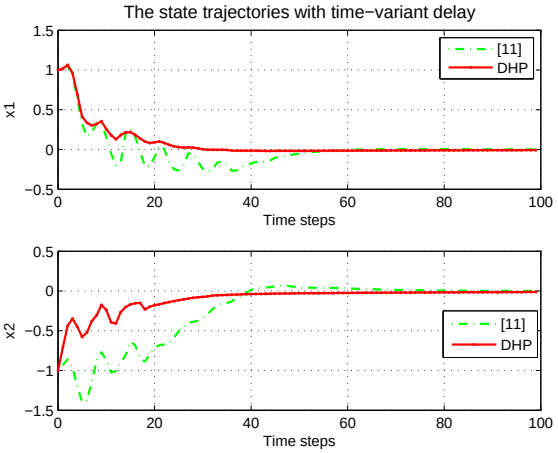


Figure 9: The state trajectories x with time-variant delay.

the DHP algorithm shows improved performance, while the feedback controller in [11] cannot guarantee stability with such a large uncertainty.

We can make conclusion from the above results that the iterative DHP algorithm presented in this paper overcomes the effect of time delay perfectly and its robustness is very good.

6. Conclusions

The paper proposes a novel iterative algorithm for optimally controlling systems represented by a large class of nonlinear discrete-time systems affected by an unknown time variant delay and system uncertainties. The iterative DHP algorithm has been envisaged to design the optimal controller and was shown to converge to the optimal controller. Three feedforward neural networks have been considered to ap-proximate the key elements required by the DHP, namely the performance index function, the optimal control policy and the system, respectively. Simulation results show the improved performance of the proposed optimal control approach.

7. Acknowledgments

We acknowledge Dr. Ding Wang and Qinglai Wei for their valuable discussions.

[1] M. Z. Manu, J. Mohammad, Time-Delay Systems: Analysis, Optimization and Applications, New York, U. S. A.: North-Holland, 1987.

[2] S. I. Niculescu, Delay Effects on Stability: a Robust Control Approach. Springer, New York, 2001.

[3] M. Fliess, R. Marquez, H. Mounier, An extension of predictive control, PID regulators and smith predictors to some linear delay systems. International Journal of Control, 75 (10) (2002) 728-743.

[4] Z. Shafiei, A. T. Shenton, Frequency-domain design of PID controllers for stable and unst-able systems with time delay, Automatica, 33 (12) (1997) 2223-2232.

[5] J. Shen, B. Chen, F. Kung, Memoryless stabilization of uncertain dynamic delay systems: Riccati equation approach, IEEE Transactions on Automatic Control, 36 (1991) 638-640.

[6] S. K. Nguang, Robust stabilization of a class of time-delay nonlinear systems, IEEE Transactions on Automatic Control, 45(4) (2000)756-762.

[7] F. E. Sarabi, H. Khatibi, Robust Stability Analysis and Synthesis of Linear Time-Delay Systems via LMIs, 2010 49th IEEE Conference on Decision and Control, 2010, pp. 615-620.

[8] N. Xie, G. Y. Tang, P. Liu, Optimal guaranteed cost control for nonlinear discrete-time uncertain systems with state delay, 2004 5th World Congress on Intelligent Control and Automation(WCICA), 2004, pp. 893-896.

[9] H. Zhang, S. Lun, D. Liu, Fuzzy H Filter Design for a Class of Nonlinear Discrete-Time Systems With Multiple Time Delays, IEEE Transactions on Fuzzy Systems, 15(3) (2007) 453-469.

[10] S. Y. Han, G. Y. Tang, Optimal tracking control for discrete-time systems with delayed state and input, 2010 8th IEEE International Conference on Control and Automation (ICCA), 2010, pp.1694-1698.

[11] Z. Gao, S. X. Ding, State and Disturbance Estimator for Time-Delay Systems With Application to Fault Estimation and Signal Compensation, IEEE Transactions on Signal Processing, 55(12) (2007) 5541-5551.

[12] Q.L. Wei, H.G. Zhang, D.R. Liu, Y. Zhao, An optimal control scheme for a class of discrete-time nonlinear systems with time delays using adaptive dynamic programming, Acta Automatica Sinica 36 (2010) 121-129.

[13] P. J. Werbos, Advanced forecasting methods for global crisis warning and models of intel-ligence, General Systems Yearbook, (22) (1977) 25-38.

[14] P. J. Werbos, Approximate dynamic programming for real-time control and neural modeling, in Handbook of Intelligent Control: Neural, Fuzzy, and Adaptive Approaches, D.A. White and D.A. Sofge, Ed., New York: Van Nostrand Reinhold, (13)1992.

[15] J. J. Murray, C. J. Cox, G. G. Lendaris, R. Saeks, Adaptive dynamic programming, IEEE Transactions on System, Man, Cybernetics. C, Appl. Rev., 32(2) (2002) 140-153.

[16] F. Y. Wang, H. Zhang, D. Liu, Adaptive dynamic programming: an introduction, IEEE Computational Intelligence Magazine, 4(2) (2009) 39-47.

[17] D. Liu, Q. Wei, Adaptive dynamic programming for a class of discrete-time non-affine nonlinear systems with time-delays, The 2010 International Joint Conference on Neural Networks (IJCNN), 2010, pp.1-6.

[18] R. Song, H.g Zhang, Y. Luo, Q. Wei, Optimal control laws for time-delay systems with saturating actuators based on heuristic dynamic programming, Neurocomputing, 73(16) (2010) 3020-3027.

[19] H. Zhang, R. Song, Q. Wei, T. Zhang, Optimal Tracking Control for a Class of Nonlinear Discrete-Time Systems With Time Delays Based on Heuristic Dynamic Programming, IEEE Transactions on Neural Networks, 22(12) (2011) 1851-1862.

[20] R.Z. Song, D.S. Yang, H.G. Zhang, Near-optimal control laws based on heuristic dynamic programming iteration algorithm, in: International Conference on Networking, Sensing and Control, 2010, pp. 261-266.

[21] D. Zhao, X. Bai, F. Wang, J. Xu, W. Yu, DHP for coordinated freeway ramp metering, IEEE Transactions on Intelligent Transportation Systems, 12 (4) (2011) 990-999.

[22] D. Zhao, Z. Zhang, Y. Dai. Self-teaching adaptive dynamic programming for Go-Moku, Neurocomputing, 78 (1) (2012) 23-29.

[23] D. Wang, D. Liu, D. Zhao, Y. Huang, D. Zhang. A neural-network-based iterative GDHP approach for solving a class of nonlinear optimal control problems with control constraints. Neural Computing and Applications, 2(22) (2011) 219-227.

[24] D. Vrabie, F. Lewis, Neural network approach to continuous-time direct adaptive optimal control for partially unknown nonlinear systems, Neural Networks, 22 (3) (2009) 237-246.

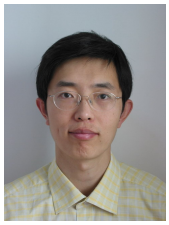
[25] C. C. Hua, X. P Guan, P. Shi, Robust Backstepping Control for a Class of Time Delay Systems, IEEE Transactions on Automatic Control, 50 (6) (2005) 894-899.

[26] A. Al-Tamimi, F.L. Lewis, Discrete-time nonlinear HJB solution using approximate dynamic programming: convergence proof, in: IEEE International Symposium on Approximate Dynamic Programming and Reinforcement Learning, 2007.

[27] A. Al-Tamimi, F. L. Lewis, M. Abu-Khalaf, Discrete-time nonlinear HJB solution using approximate dynamic programming: convergence proof, IEEE Transactions on System, Man, and Cybernetics, B, Cybernetics, 38(4) (2008) 943-949.

[28] M. Abu-Khalaf, F. L. Lewis, Nearly optimal control laws for nonlinear systems with saturating actuators using a neural network HJB approach, Automatica, 41(5) (2005) 779-791.

Bin Wang received B.S. degree in China University of Petroleum, Dongying, China in 2006. He is currently working towards Ph.D. degree at the State Key Laboratory of management and Control for Complex Systems, Institute of Automation, Chinese Academy of Sciences, China. His current research interests include neural-networks-based control, approximate dynamic programming and their industrial application.



Dongbin Zhao received the B.S., M.S., Ph.D. degrees in Aug. 1994, Aug. 1996, and Apr. 2000 respectively, in materials processing engineering from Harbin Institute of Technology, China. Dr. Zhao was a postdoctoral fellow in humanoid robot at the Department of Mechanical Engineering, Tsinghua University, China, from May 2000 to Jan. 2002.

He is currently a professor at the State Key Laboratory of Management and Control for Complex Systems, Institute of Automation, Chinese Academy of Sciences, China. He has published one book and over thirty international journal papers. His current research interests lies in the area of computational intelligence, adaptive dynamic programming, robotics, intelligent transportation systems, and process simulation.

Dr. Zhao is an Associate Editor of the IEEE Transactions on Neural Networks and Learning Systems, and Cognitive Computation.

Cesare Alippi received the degree in electronic engineering cum laude in 1990 and the PhD in 1995 from Politecnico di Milano, Italy. Currently, he is a Full Professor of information processing systems with the Politecnico di Milano. He has been a visiting researcher at UCL (UK), MIT (USA), ESPCI (F), CASIA (CN). Alippi is an IEEE Fellow, Vice-President education of the IEEE Computational Intelligence Society (CIS), Associate editor (AE) of the IEEE Computational Intelligence Magazine, past AE of the IEEE-Tran. Neural Networks, IEEE-Trans Instrumentation and Measurements (2003-09) and member and chair of other IEEE committees including the IEEE Rosenblatt award.

In 2004 he received the IEEE Instrumentation and Measurement Society Young Engineer Award; in 2011 has been awarded Knight of the Order of Merit of the Italian Republic.Current research activity addresses adaptation and learning in non-stationary environments and Intelligent embedded systems.

He holds 5 patents and has published about 200 papers in international journals and conference proceedings.

Derong Liu received the Ph.D. degree in electrical engineering from the University of Notre Dame in 1994. Dr. Liu was a Staff Fellow with General Motors Research and Development Center, Warren, MI, from 1993 to 1995. He was an Assistant Professor in the Department of Electrical and Computer Engineering, Stevens Institute of Technology, Hoboken, NJ, from 1995 to 1999. He joined the University of Illinois at Chicago in 1999, and became a Full Professor of electrical and computer engineering and of computer science in 2006. He was selected for the "100 Talents Program" by the Chinese Academy of Sciences in 2008. He has published 10 books. Dr. Liu has been an Associate Editor of several IEEE publications. Currently, he is the Editor-in-Chief of the IEEE Transactions on Neural Networks and Learning Systems, and an Associate Editor of the IEEE Transactions on Control Systems Technology. He was an elected AdCom member of the IEEE Computational Intelligence Society (2006-2008). He received the Faculty Early Career Development (CAREER) award from the National Science Foundation (1999), the University Scholar Award from University of Illinois (2006-2009), and the Overseas Outstanding Young Scholar Award from the National Natural Science Foundation of China (2008).