Soft Comput (2016) 20:697–706
DOI 10.1007/s00500-014-1533-0

METHODOLOGIES AND APPLICATION

Neuro-optimal tracking control for a class of discrete-time nonlinear systems via generalized value iteration adaptive dynamic programming approach

Qinglai Wei · Derong Liu · Yancai Xu

Published online: 25 November 2014

Abstract In this paper, a novel value iteration adaptive dynamic programming (ADP) algorithm, called "generalized value iteration ADP" algorithm, is developed to solve infinite horizon optimal tracking control problems for a class of discrete-time nonlinear systems. The developed generalized value iteration ADP algorithm permits an arbitrary positive semi-definite function to initialize it, which overcomes the disadvantage of traditional value iteration algorithms. Convergence property is developed to guarantee that the iterative performance index function will converge to the optimum. Neural networks are used to approximate the iterative performance index function and compute the iterative control policy, respectively, to implement the iterative ADP algorithm. Finally, a simulation example is given to illustrate the performance of the developed algorithm.

Keywords Adaptive dynamic programming · Approximate dynamic programming · Adaptive critic designs · Optimal control · Neural networks · Nonlinear systems · Reinforcement learning

Communicated by V. Loia.

Q. Wei (✉) · D. Liu · Y. Xu
The State Key Laboratory of Management and Control for Complex Systems, Institute of Automation, Chinese Academy of Sciences, Beijing 100190, China
e-mail: qinglai.wei@ia.ac.cn

D. Liu
e-mail: derong.liu@ia.ac.cn

Y. Xu
e-mail: yancai.xu@ia.ac.cn

1 Introduction

Optimal tracking control of nonlinear systems has always been the key focus in the control field in the latest several decades (Rugh 1971; Mohler and Kolodziej 1981; Biswas et al. 2014; Fortier et al. 2014; Rubio 2014). Although dynamic programming is a powerful tool to solve the optimization and optimal control problems for nonlinear systems (Kundu et al. 2014; Kouramas et al. 2013; Chang 2013), it is often computationally untenable to run true dynamic programming algorithms, i.e., as a result of the well-known "curse of dimensionality" (Bellman 1957). Adaptive dynamic programming (ADP), proposed by Werbos (1977, 1991), is an effective adaptive learning control approach to solve optimal control problems forward-in-time (Liu et al. 2005, 2008; Ni and He 2013; Prokhorov and Wunsch 1997; Wang et al. 2011; Xu and Jagannathan 2013; Heydari and Balakrishnan 2013; Wei et al. 2014a). There are several synonyms used for ADP including "adaptive critic designs" (Werbos 1991; Al-Tamimi et al. 2007), "adaptive dynamic programming" (Murray et al. 2002; Wang et al. 2009), "approximate dynamic programming" (Werbos 1992; Liu and Wei 2014a), "neural dynamic programming" (Enns and Si 2003), "neuro-dynamic programming" (Bertsekas and Tsitsiklis 1996), and "reinforcement learning" (Si and Wang 2001; Sutton and Barto 1998). Iterative methods are important methods in ADP to obtain the optimal control law iteratively and have received lots of attention (Bhasin et al. 2013; Liu and Wei 2014a; Song et al. 2013, 2014; Wei and Liu 2012, 2013; Wei et al. 2013; Wei and Liu 2014b, c, d; Wei et al. 2014b; Zhang et al. 2013, 2014).

There are two main iterative ADP algorithms which are based on policy and value iterations (Lewis et al. 2012). Policy iteration algorithms for optimal control of continuous-time systems were given in Abu-Khalaf and Lewis (2005),

Murray et al. (2002), Zhang et al. (2011). In Liu and Wei (2014b), policy iteration algorithm for discrete-time systems was developed. Value iteration algorithms are a class of the most important iterative ADP algorithms (Al-Tamimi et al. 2008; Bertsekas 2007; Powell 2007; Wei et al. 2009; Wei and Liu 2014a, b). Value iteration algorithms of ADP were given in Bertsekas and Tsitsiklis (1996), where the initial admissible control law of the policy iteration algorithm is avoided. In Al-Tamimi et al. (2008) and Lincoln and Rantzer (2006), starting from a zero initial performance index function, it is proven that the iterative performance index function is a non-decreasing sequence and converges to the optimum. In 2008, Zhang et al. applied value iteration algorithm to solve optimal tracking control problems for nonlinear systems. In Zhang et al. (2009), value iteration of ADP, which was referred to as dual heuristic dynamic programming (DHP), was implemented using RBF neural networks.

However, in the previous value iterations, to guarantee the monotonicity of value iteration algorithms, nearly all the traditional value iteration algorithms are required to start from a zero initial condition. These value iteration algorithms are called "traditional value iteration algorithms." To the best of our knowledge, there are no discussions on the value iteration algorithms with non-zero initial conditions. This motivates our research.

In this paper, inspired by Lincoln and Rantzer (2006), Liu and Wei (2013), Zhang et al. (2008), a new value iteration ADP algorithm, called "generalized value iteration algorithm," is developed for discrete-time nonlinear systems. First, it will show that the developed value iteration algorithm permits to start from an arbitrary positive semi-definite function. By system transformations, the optimal tracking problem is transformed into an optimal regulation problem. Next, the convergence properties of the developed value iteration algorithm are presented to guarantee that the iterative performance index function is convergent to the optimum. Implementing the algorithms by neural networks, the effectiveness of the developed algorithm will be justified by simulation results.

This paper is organized as follows: In Sect. 2, the problem statement is presented. In Sect. 3, the generalized value iteration algorithm will be derived. The convergence properties will also be analyzed in this section. In Sect. 4, the neural network implementation for the optimal control scheme is discussed. In Sect. 5, a simulation example is given to demonstrate the effectiveness of the proposed algorithm. Finally, in Sect. 6, the conclusion is drawn.

2 Problem statement

Consider the following class of nonlinear systems with the form

$$x_{k+1} = F(x_k, u_k), \tag{1}$$

where $x_k \in \Re^n$ is the system state and $u_k \in \Re^m$ is the control input. For infinite-time optimal tracking problem, the objective is to design optimal control $u(x_k)$ for system (1) such that the state x_k tracks the specified desired trajectory $\eta_k \in \Re^n, k = 0, 1, \ldots$. In this paper, we assume that there exists a feedback control $u_{e,k}$, which satisfies the following equation:

$$\eta_k = F(\eta_k, u_{e,k}), \tag{2}$$

where $u_{e,k}$ is called the desired control.

Remark 1 It should be pointed out that for a large class of nonlinear systems, there exists a feedback control $u_{e,k}$ that satisfies (2). For example, for all the affine nonlinear systems with expression

$$x_{k+1} = f(x_k) + g(x_k)u_k, \tag{3}$$

where $g(x_k)$ is invertible, the desired control $u_{e,k}$ can be expressed as

$$u_{e,k} = g^{-1}(\eta_k)(\eta_{k+1} - f(\eta_k)), \tag{4}$$

where $g^{-1}(\eta_k)g(\eta_k) = I$ and $I \in \Re^{m\times m}$ is the identity matrix. Hence, the investigation of the optimal control problem for system (1) is valuable.

Define the tracking error as

$$z_k = x_k - \eta_k. \tag{5}$$

Define the following quadratic performance index

$$J(z_0, \underline{u}_0) = \sum_{k=0}^{\infty} \left\{ z_k^T Q z_k + (u_k - u_{e,k})^T R (u_k - u_{e,k}) \right\}, \tag{6}$$

where $Q \in \Re^{n\times n}$ and $R \in \Re^{m\times m}$ are positive definite matrices and $\underline{u}_0 = (v_0, v_1, \ldots)$. Let

$$U(z_k, v_k) = z_k^T Q z_k + v_k^T R v_k \tag{7}$$

be the utility function, where $v_k = u_k - u_{e,k}$ and $u_{e,k}$ is the desired control that satisfies (2).

We will study optimal tracking control problems for (1). The goal of this paper is to find an optimal tracking control scheme which tracks the desired trajectory η_k and simultaneously minimizes the performance index function (6). The optimal performance index function is defined as

$$J^*(z_k) = \inf_{\underline{v}_k} \left\{ J(z_k, \underline{v}_k) \right\}, \tag{8}$$

where $\underline{v}_k = (v_k, v_{k+1}, \ldots)$. According to Bellman's principle of optimality, $J^*(z_k)$ satisfies the discrete-time Hamilton–Jacobi–Bellman (HJB) equation

$$J^*(z_k) = \inf_{v_k} \left\{ U(z_k, v_k) + J^*(F(z_k, v_k)) \right\}. \tag{9}$$

Then, the optimal control law is expressed as

$$v^*(z_k) = \arg\inf_{v_k} \left\{ U(z_k, v_k) + J^*(z_{k+1}) \right\}. \tag{10}$$

Hence, the HJB Eq. (9) can be written as

$$J^*(z_k) = U(z_k, v^*(z_k)) + J^*(z_{k+1}). \tag{11}$$

Generally, $J^*(z_k)$ is a high nonlinear and non-analytic function, which cannot be obtained by directly solving the HJB equation (11). In this paper, a new generalized value iteration is developed to obtain $J^*(z_k)$ iteratively with new convergence analysis.

3 Generalized value iteration algorithm of ADP for optimal tracking problems

In this section, a new generalized value iteration algorithm is developed to obtain the optimal tracking control law for nonlinear systems (1). The goal of the present iterative ADP algorithm is to construct an optimal control law $u^*(z_k)$, $k = 0, 1, \ldots$, which moves an arbitrary initial state x_0 to the desired trajectory η_k, and simultaneously minimizes the performance index function. Convergence property will be analyzed to guarantee that the performance index functions converge to the optimum.

3.1 Derivation of the generalized value iteration algorithm

In the developed generalized value iteration algorithm, the performance index function and control law are updated by iterations, with the iteration index i increasing from 0 to infinity.

For $\forall z_k \in \Re^n$, let the initial function $\Psi(z_k)$ be an arbitrary positive semi-definite function. Then, let the initial performance index function

$$V_0(z_k) = \Psi(z_k), \tag{12}$$

and the iterative control law $v_0(z_k)$ can be computed as follows:

$$\begin{aligned} v_0(z_k) &= \arg\min_{v_k} \{U(z_k, v_k) + V_0(z_{k+1})\} \\ &= \arg\min_{v_k} \{U(z_k, v_k) + V_0(F(z_k, v_k))\}, \end{aligned} \tag{13}$$

where $V_0(z_{k+1}) = \Psi(z_{k+1})$. The performance index function can be updated as

$$V_1(z_k) = U(z_k, v_0(z_k)) + V_0(F(z_k, v_0(z_k)). \tag{14}$$

For $i = 1, 2, \ldots$, the iterative ADP algorithm will iterate between

$$\begin{aligned} v_i(z_k) &= \arg\min_{v_k} \{U(z_k, v_k) + V_i(z_{k+1})\} \\ &= \arg\min_{v_k} \{U(z_k, v_k) + V_i(F(z_k, v_k))\}, \end{aligned} \tag{15}$$

and

$$\begin{aligned} V_{i+1}(z_k) &= \min_{v_k} \{U(z_k, v_k) + V_i(z_{k+1})\} \\ &= U(z_k, v_i(z_k)) + V_i(F(z_k, v_i(z_k)). \end{aligned} \tag{16}$$

Remark 2 In the traditional value iteration algorithms, such as Al-Tamimi et al. (2007, 2008), Bertsekas (2007), Lincoln and Rantzer (2006), Powell (2007), Zhang et al. (2008, 2009), the initial performance index function is required to be zero. From (12)–(16), the generalized value iteration ADP algorithm permits an arbitrary positive semi-definite function to initialize it. Hence, we can say that the traditional value iteration algorithms are just special cases of the developed generalized value iteration algorithm.

From the generalized iterative ADP algorithm (12)–(16), we can see that the iterative performance index function $V_i(z_k)$ is used to approximate $J^*(z_k)$ and the iterative control law $v_i(z_k)$ is used to approximate $u^*(z_k)$. Therefore, when $i \to \infty$, the algorithm should be convergent which makes $V_i(z_k)$ and $v_i(z_k)$ converge to the optimal ones. In the next subsection, we will show the properties of the generalized value iterative ADP algorithm.

3.2 Properties of the generalized value iteration algorithm

In Al-Tamimi et al. (2008), Lincoln and Rantzer (2006), for zero initial performance index function, it was proven that the iterative performance index function is monotonically non-decreasing and converges to the optimum. However, for arbitrary positive semi-definite initial functions, the analysis method for the traditional value iteration algorithms is invalid. In Lincoln and Rantzer (2006), Liu and Wei (2013), the upper bound of the iterative performance index function were use to analyze the convergence of their algorithms instead of analyzing the value of iterative performance index function. In this paper, inspired by Lincoln and Rantzer (2006), Liu and Wei (2013), new convergence analysis for the generalized value iteration algorithm is developed in this section.

Theorem 1 *For $i = 0, 1, \ldots$, let $V_i(z_k)$ and $v_i(z_k)$ be obtained by (12)–(16). Let $\underline{\varrho}$, $\overline{\varrho}$, $\underline{\varsigma}$ and $\overline{\varsigma}$ be constants that satisfy*

$$0 < \underline{\varrho} \le \overline{\varrho} < \infty \tag{17}$$

and

$$0 \le \underline{\varsigma} \le \overline{\varsigma} < 1, \tag{18}$$

respectively. If for $\forall z_k$, the constants $\underline{\varrho}$, $\overline{\varrho}$, $\underline{\varsigma}$ and $\overline{\varsigma}$ satisfy

$$\underline{\varrho} U(z_k, v_k) \le J^*(F(z_k, v_k)) \le \overline{\varrho} U(z_k, v_k) \tag{19}$$

and

$$\underline{\varsigma} J^*(z_k) \le V_0(z_k) \le \overline{\varsigma} J^*(z_k) \tag{20}$$

uniformly, we have the iterative performance index function $V_i(z_k)$ satisfies

$$\left(1 + \frac{\underline{\varsigma} - 1}{(1 + \overline{\varrho}^{-1})^i}\right) J^*(z_k) \le V_i(z_k) \le \left(1 + \frac{\overline{\varsigma} - 1}{(1 + \underline{\varrho}^{-1})^i}\right) J^*(z_k). \tag{21}$$

Proof The theorem can be proved in two steps.

(1) Prove the left side of the inequality (21).

Mathematical induction is employed to prove the conclusion. Let $i = 1$. We have

$$\begin{aligned} V_1(z_k) &= \min_{v_k} \{U(z_k, v_k) + V_0(z_{k+1})\} \\ &\ge \min_{v_k} \left\{U(z_k, v_k) + \underline{\varsigma} J^*(z_{k+1})\right\} \\ &\ge \min_{v_k} \left\{\left(1 + \overline{\varrho}\frac{\underline{\varsigma} - 1}{1 + \overline{\varrho}}\right) U(z_k, v_k) + \left(\underline{\varsigma} - \frac{\underline{\varsigma} - 1}{1 + \overline{\varrho}}\right) J^*(z_{k+1})\right\} \\ &\ge \left(1 + \frac{\underline{\varsigma} - 1}{(1 + \overline{\varrho}^{-1})}\right) \min_{v_k} \left\{U(z_k, v_k) + J^*(z_{k+1})\right\} \\ &= \left(1 + \frac{\underline{\varsigma} - 1}{(1 + \overline{\varrho}^{-1})}\right) J^*(z_k). \end{aligned} \tag{22}$$

Assume the conclusion holds for $i = l - 1$, $l = 1, 2, \ldots$. Then for $i = l$, we have

$$\begin{aligned} V_{l+1}(z_k) &= \min_{v_k} \{U(z_k, v_k) + V_l(z_{k+1})\} \\ &\ge \min_{v_k} \left\{U(z_k, v_k) + \left(1 + \frac{\underline{\varsigma} - 1}{(1 + \overline{\varrho}^{-1})^{l-1}}\right) J^*(z_{k+1})\right\} \\ &\ge \left(1 + \frac{\underline{\varsigma} - 1}{(1 + \overline{\varrho}^{-1})^{l}}\right) \min_{v_k} \left\{U(z_k, v_k) + J^*(z_{k+1})\right\} \\ &= \left(1 + \frac{\underline{\varsigma} - 1}{(1 + \overline{\varrho}^{-1})^{l}}\right) J^*(z_k). \end{aligned} \tag{23}$$

(2) Prove the right side of the inequality (21).

We also use mathematical induction to prove the conclusion. Let $i = 1$. We have

$$\begin{aligned} V_1(z_k) &= \min_{v_k} \{U(z_k, v_k) + V_0(z_{k+1})\} \\ &\le \min_{v_k} \left\{U(z_k, v_k) + \overline{\varsigma} J^*(z_{k+1})\right\} \\ &\le \min_{v_k} \left\{U(z_k, v_k) + \overline{\varsigma} J^*(z_{k+1}) - \frac{\overline{\varsigma} - 1}{(1 + \underline{\varrho})} \left(J^*(z_{k+1}) - \underline{\varrho} U(z_k, v_k)\right)\right\} \\ &\le \left(1 + \frac{\overline{\varsigma} - 1}{(1 + \underline{\varrho}^{-1})}\right) \min_{v_k} \left\{U(z_k, v_k) + J^*(z_{k+1})\right\} \\ &= \left(1 + \frac{\overline{\varsigma} - 1}{(1 + \underline{\varrho}^{-1})}\right) J^*(z_k). \end{aligned} \tag{24}$$

Assume that the conclusion holds for $i = l - 1, l = 1, 2, \ldots$. Then for $i = l$, we have

$$\begin{aligned} V_{l+1}(z_k) &= \min_{v_k} \{U(z_k, v_k) + V_l(z_{k+1})\} \\ &\le \min_{v_k} \left\{U(z_k, v_k) + \left(1 + \frac{\overline{\varsigma} - 1}{(1 + \underline{\varrho}^{-1})^{l-1}}\right) J^*(z_{k+1})\right\} \\ &\le \left(1 + \frac{\overline{\varsigma} - 1}{(1 + \underline{\varrho}^{-1})^{l}}\right) \min_{v_k} \left\{U(z_k, v_k) + J^*(z_{k+1})\right\} \\ &= \left(1 + \frac{\overline{\varsigma} - 1}{(1 + \underline{\varrho}^{-1})^{l}}\right) J^*(z_k). \end{aligned} \tag{25}$$

The proof is completed. □

Theorem 2 *For $i = 0, 1, \ldots$, let $V_i(z_k)$ and $v_i(z_k)$ be obtained by (12)–(16). Let $\underline{\varrho}$, $\overline{\varrho}$, $\underline{\varsigma}$ and $\overline{\varsigma}$ be constants that satisfy (17) and*

$$0 \le \underline{\varsigma} \le 1 \le \overline{\varsigma} < \infty. \tag{26}$$

If for $\forall z_k$, the constants $\underline{\varrho}$, $\overline{\varrho}$, $\underline{\varsigma}$, and $\overline{\varsigma}$ make (19) and (20) hold uniformly. Then, we have the iterative performance index

function $V_i(z_k)$ satisfies

$$\left(1+\frac{\underline{\varsigma}-1}{(1+\overline{\varrho}^{-1})^i}\right)J^*(z_k)\le V_i(z_k)\le\left(1+\frac{\overline{\varsigma}-1}{(1+\overline{\varrho}^{-1})^i}\right)J^*(z_k). \tag{27}$$

Proof The left side of inequality (27) can be proven according to (22) and (23). Now we prove the right side of inequality (27) by mathematical induction. Let $i=0$. We have

$$\begin{aligned}V_1(z_k)&=\min_{v_k}\{U(z_k,v_k)+V_0(F(z_k,v_k))\}\\&\le\min_{v_k}\left\{U(z_k,v_k)+\overline{\varsigma}J^*(F(z_k,v_k))+\frac{\overline{\varsigma}-1}{(1+\overline{\varrho})}\left(\overline{\varrho}U(z_k,v_k)-J^*(F(z_k,v_k))\right)\right\}\\&\le\left(1+\frac{\overline{\varsigma}-1}{(1+\overline{\varrho}^{-1})}\right)J^*(z_k).\end{aligned} \tag{28}$$

Assume that the conclusion holds for $i=l-1, l=1,2,\ldots$. Then for $i=l$, we have

$$\begin{aligned}V_{l+1}(z_k)&=\min_{v_k}\{U(z_k,v_k)+V_l(F(z_k,v_k))\}\\&\le\min_{v_k}\left\{U(z_k,v_k)+\left(1+\frac{\overline{\varsigma}-1}{(1+\overline{\varrho}^{-1})^{l-1}}\right)J^*(F(z_k,v_k))\right\}\\&\le\left(1+\frac{\overline{\varsigma}-1}{(1+\overline{\varrho}^{-1})^{l}}\right)J^*(z_k).\end{aligned} \tag{29}$$

The proof is completed. □

Theorem 3 *For $i=0,1,\ldots$, let $V_i(z_k)$ and $v_i(z_k)$ be obtained by (12)–(16). Let $\underline{\varrho}$, $\overline{\varrho}$, $\underline{\varsigma}$ and $\overline{\varsigma}$ be constants that satisfy (17) and*

$$1\le\underline{\varsigma}\le\overline{\varsigma}<\infty, \tag{30}$$

respectively. If for $\forall z_k$, the constants $\underline{\varrho}$, $\overline{\varrho}$, $\underline{\varsigma}$ and $\overline{\varsigma}$ make (19) and (20) hold uniformly, then we have the iterative performance index function $V_i(z_k)$ satisfies (21).

Theorem 4 *For $i=0,1,\ldots$, let $V_i(z_k)$ and $v_i(z_k)$ be obtained by (12)–(16). Let $\underline{\varrho}$, $\overline{\varrho}$, $\underline{\varsigma}$ and $\overline{\varsigma}$ be constants that satisfy (17) and*

$$0\le\underline{\varsigma}\le\overline{\varsigma}<\infty, \tag{31}$$

respectively. If for $\forall z_k$, the constants $\underline{\varrho}$, $\overline{\varrho}$, $\underline{\varsigma}$ and $\overline{\varsigma}$ make (19) and (20) hold uniformly, then we have the iterative performance index function $V_i(z_k)$ convergent to the optimal performance index function $J^*(z_k)$, i.e.,

$$\lim_{i\to\infty}V_i(z_k)=J^*(z_k). \tag{32}$$

Proof According to (21) and (27), respectively, let $i\to\infty$ and we can obtain

$$\begin{aligned}&\lim_{i\to\infty}\left\{\left(1+\frac{\underline{\varsigma}-1}{(1+\overline{\varrho}^{-1})^i}\right)J^*(z_k)\right\}\\&=\lim_{i\to\infty}\left\{\left(1+\frac{\overline{\varsigma}-1}{(1+\underline{\varrho}^{-1})^i}\right)J^*(z_k)\right\}\\&=\lim_{i\to\infty}\left\{\left(1+\frac{\overline{\varsigma}-1}{(1+\overline{\varrho}^{-1})^i}\right)J^*(z_k)\right\}\\&=J^*(z_k).\end{aligned} \tag{33}$$

The proof is completed. □

Remark 3 From Theorem 4, we can see that the iterative performance index function will converge to the optimum as $i\to\infty$, which is independent from the initial performance index function $\Psi(z_k)$. Furthermore, for arbitrary constants $\underline{\varrho}$, $\overline{\varrho}$, $\underline{\varsigma}$ and $\overline{\varsigma}$ that satisfy (17) and (30), respectively, the iterative performance index function $V_i(z_k)$ can be guaranteed to converge to the optimum as $i\to\infty$. Hence the estimation of the constants $\underline{\varrho}$, $\overline{\varrho}$, $\underline{\varsigma}$ and $\overline{\varsigma}$ can be avoided.

Corollary 1 *For $i=0,1,\ldots$, let $V_i(z_k)$ and $v_i(z_k)$ be obtained by (12)–(16). If for $\forall z_k$, the initial performance index function $\Psi(z_k)\le J^*(z_k)$, then for $\forall i\ge 0$, we have*

$$V_i(z_k)\le J^*(z_k) \tag{34}$$

holds.

Proof The statement can be proven by mathematical induction. The conclusion holds obviously for $i=0$. For $i=1$, as $\Psi(z_k)\le J^*(z_k)$, we have

$$\begin{aligned}V_1(z_k)&=U(z_k,v_0(z_k))+\Psi(z_{k+1})\\&=\min_{v_k}\{U(z_k,v_k)+\Psi(z_{k+1})\}\\&\le\min_{v_k}\left\{U(z_k,v_k)+J^*(z_{k+1})\right\}\\&=J^*(z_k).\end{aligned} \tag{35}$$

Assume that the conclusion holds for $i=l-1, l=1,2,\ldots$. Then, for $i=l$, we have

$$\begin{aligned} V_{l+1}(z_k) &= U(z_k, v_l(z_k)) + V_l(z_{k+1}) \\ &= \min_{v_k} \{U(z_k, v_k) + V_l(z_{k+1})\} \\ &\le \min_{v_k} \left\{U(z_k, v_k) + J^*(z_{k+1})\right\} \\ &= J^*(z_k), \end{aligned} \tag{36}$$

which complete the proof. □

Corollary 2 *For $i = 0, 1, \ldots$, let $V_i(z_k)$ and $v_i(z_k)$ be obtained by (12)–(16). If for $\forall z_k$, the initial performance index function $\Psi(z_k) \ge J^*(z_k)$, then for $\forall i \ge 0$, we have $V_i(z_k) \ge J^*(z_k)$ holds.*

Theorem 5 *For $i = 0, 1, \ldots$, let $V_i(z_k)$ and $v_i(z_k)$ be obtained by (12)–(16). If for $\forall z_k \in \Re^n$, we have*

$$V_1(z_k) \le V_0(z_k) \tag{37}$$

holds, where $V_0(z_k)$ is expressed by (12), then the iterative performance index function $V_i(z_k)$ is a monotonically non-increasing sequence for $\forall\, i \ge 0$, i.e.,

$$V_{i+1}(z_k) \le V_i(z_k). \tag{38}$$

Proof We prove this by mathematical induction. First, we let $i = 1$. According to (16) and (37), we have

$$\begin{aligned} V_2(z_k) &= \min_{v_k} \{U(z_k, v_k) + V_1(z_{k+1})\} \\ &\le \min_{v_k} \{U(z_k, v_k) + V_0(z_{k+1})\} \\ &= V_1(z_k). \end{aligned} \tag{39}$$

Assume the conclusion holds for $i = l - 1, l = 1, 2, \ldots$; then for $i = l$ we have

$$\begin{aligned} V_{l+1}(z_k) &= \min_{v_k} \{U(z_k, v_k) + V_l(z_{k+1})\} \\ &\le \min_{v_k} \{U(z_k, v_k) + V_{l-1}(z_{k+1})\} \\ &= V_l(z_k). \end{aligned} \tag{40}$$

The proof is completed. □

Theorem 6 *For $i = 0, 1, \ldots$, let $V_i(z_k)$ and $v_i(z_k)$ be obtained by (12)–(16). If for $\forall z_k \in \Re^n$, we have*

$$V_1(z_k) \ge V_0(z_k) \tag{41}$$

holds, where $V_0(z_k)$ is expressed by (12), then the iterative performance index function $V_i(z_k)$ is a monotonically non-decreasing sequence for $\forall\, i \ge 0$, i.e.,

$$V_{i+1}(z_k) \ge V_i(z_k) \tag{42}$$

for $\forall\, i \ge 1$.

Remark 4 If for $\forall z_k \in \Re^n$, let the initial performance index function $V_0(z_k) \equiv 0$, and then the generalized value iteration algorithm is reduced to the traditional value iteration algorithms in Al-Tamimi et al. (2008), Zhang et al. (2008). In the traditional value iteration algorithms, the iterative performance index function is monotonically non-decreasing and converges to the optimum. In the generalized value iteration algorithm, the iterative performance index function can be monotonically non-increasing, non-decreasing and non-monotonically converge to the optimum. So, we can say that the convergence property of the traditional value iteration algorithms is a special case of the generalized value iteration algorithm.

4 Implementation of the generalized value iteration algorithm by neural networks

In this section, three-layer back-propagation (BP) neural networks are introduced to approximate $V_i(z_k)$ and compute the control law $v_i(z_k)$, respectively. Assume that the number of hidden layer neurons is denoted by l. The weight matrix between the input layer and hidden layer is denoted by Y, and the weight matrix between the hidden layer and output layer is denoted by W. Then the output of three-layer neural network is represented by

$$\hat{F}(X, Y, W) = W^T \sigma(Y^T X), \tag{43}$$

where $\sigma(Y^T X) \in R^l, [\sigma(\mathbb{z})]_i = \dfrac{e^{\ i} - e^{-\ i}}{e^{\ i} + e^{-\ i}}, i = 1, \ldots l,$ are the activation function.

There are two networks, which are critic and action networks, respectively, to implement the generalized value iteration algorithm. The whole structure diagram is shown in Fig. 1.

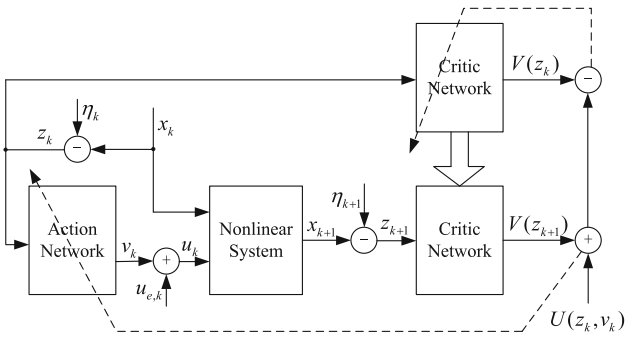

Fig. 1 The structure diagram of the algorithm

4.1 The critic network

The critic network is used to approximate the performance index function $V_i(z_k)$. The output of the critic network is denoted as

$$\hat{V}_i(z_k) = W_{ci}^T \sigma(Y_{ci}^T z_k). \tag{44}$$

The target function can be written as

$$V_{i+1}(z_k) = U(z_k, \hat{v}_i(z_k)) + \hat{V}_i(z_{k+1}). \tag{45}$$

Then, we define the error function for the critic network as

$$e_{ci}(k) = \hat{V}_{i+1}(z_k) - V_{i+1}(z_k). \tag{46}$$

The objective function to be minimized in the critic network training is

$$E_{ci}(k) = \frac{1}{2} e_{ci}^2(k). \tag{47}$$

So the gradient-based weight update rule (Si and Wang 2001) for the critic network is given by

$$w_{c(i+1)}(k) = w_{ci}(k) + \Delta w_{ci}(k), \tag{48}$$

$$\Delta w_{ci}(k) = \alpha_c \left[-\frac{\partial E_{ci}(k)}{\partial w_{ci}(k)} \right], \tag{49}$$

$$\frac{\partial E_{ci}(k)}{\partial w_{ci}(k)} = \frac{\partial E_{ci}(k)}{\partial \hat{V}_i(z_k)} \frac{\partial \hat{V}_i(z_k)}{\partial w_{ci}(k)}, \tag{50}$$

where $\alpha_c > 0$ is the learning rate of critic network and $w_{ci}(k)$ is the weight vector of the critic network.

4.2 The action network

In the action network, the state error z_k is used as the input to create the optimal control law as the output of the network. The output can be formulated as

$$\hat{v}_i(z_k) = W_{ai}^T \sigma(Y_{ai}^T z_k). \tag{51}$$

The target of the output of the action network is given by (15). So we can define the output error of the action network as

$$e_{ai}(k) = \hat{v}_i(z_k) - v_i(z_k). \tag{52}$$

The weights of the action network are updated to minimize the following performance error measure:

$$E_{ai}(k) = \frac{1}{2} e_{ai}^T(k) e_{ai}(k). \tag{53}$$

The weight updating algorithm is similar to the one for the critic network. By the gradient descent rule, we can obtain

$$w_{a(i+1)}(k) = w_{ai}(k) + \Delta w_{ai}(k), \tag{54}$$

$$\Delta w_{ai}(k) = \beta_a \left[-\frac{\partial E_{ai}(k)}{\partial w_{ai}(k)} \right], \tag{55}$$

$$\frac{\partial E_{ai}(k)}{\partial w_{ai}(k)} = \frac{\partial E_{ai}(k)}{\partial e_{ai}(k)} \frac{\partial e_{ai}(k)}{\partial v_i(k)} \frac{\partial v_i(k)}{\partial w_{ai}(k)} \tag{56}$$

where $\beta_a > 0$ is the learning rate of action network.

The generalized value iteration algorithm implemented by action and critic networks is explained step by step and shown in Algorithm 1.

Algorithm 1 Neural network implementation for generalized value iteration.

Initialization:

1: Given a desired trajectory η_k.
2: Collect an array of system data for system (1).
3: Give a positive semi-definite function $\Psi(x_k)$.
4: Give the computation precision $\varepsilon > 0$.

Iteration:

5: According to η_k and x_k, obtain the tracking error z_k.
6: Obtain the desired control law $u_{e,k}$ by (2).
7: Let $i = 0$ and let $V_0(z_k) = \Psi(x_k)$.
8: Train the action and critic networks to obtain $v_0(z_k)$ and $V_1(z_k)$ by

$$\begin{aligned} v_0(z_k) &= \arg\min_{v_k} \{U(z_k, v_k) + V_0(z_{k+1})\} \\ &= \arg\min_{v_k} \{U(z_k, v_k) + V_0(F(z_k, v_k))\} \end{aligned}$$

and

$$V_1(z_k) = U(z_k, v_0(z_k)) + V_0(F(z_k, v_0(z_k))),$$

respectively.
9: Let $i = i + 1$.
10: Train the action and critic networks to obtain $v_i(z_k)$ and $V_{i+1}(z_k)$ by

$$\begin{aligned} v_i(z_k) &= \arg\min_{v_k} \{U(z_k, v_k) + V_i(z_{k+1})\} \\ &= \arg\min_{v_k} \{U(z_k, v_k) + V_i(F(z_k, v_k))\}, \end{aligned}$$

and

$$\begin{aligned} V_{i+1}(z_k) &= \min_{v_k} \{U(z_k, v_k) + V_i(z_{k+1})\} \\ &= U(z_k, v_i(z_k)) + V_i(F(z_k, v_i(z_k))). \end{aligned}$$

11: If $|V_{i+1}(z_k) - V_i(z_k)| \leq \varepsilon$, then goto next step. Otherwise, goto Step 9.
12: Obtain u_k^* by $u_k^* = v_i(z_k) + u_{e,k}$.
13: **return** $V_i(z_k)$ and u_k^*.

5 Simulation study

In this section, the performance of our developed algorithm will be justified by simulation results.

Example Our example is chosen as the example in Zhang et al. (2008). Consider the following nonlinear system:

$$x_{k+1} = f(x_k) + gu_k \tag{57}$$

where $x_k = [\,x_{1k}\ x_{2k}\,]^T$ and $u_k = [\,u_{1k}\ u_{2k}\,]^T$. Let the system function be expressed as

$$f(x_k) = \begin{bmatrix} 0.2x_{1k}\exp(x_{2k}^2) \\ 0.3x_{2k}^3 \end{bmatrix},\ g = \begin{bmatrix} 0.2 & 0 \\ 0 & 0.2 \end{bmatrix}.$$

The desired trajectory is set to

$$\eta_k = \left[\sin\left(k + \frac{\pi}{2}\right) \quad 0.5\cos(k)\right]^T. \tag{58}$$

According to (57) and (58), we can easily obtain the desired control

$$u_{e,k} = -\begin{bmatrix} 5 & 0 \\ 0 & 5 \end{bmatrix}\left(\eta_{k+1} - \begin{bmatrix} 0.2\eta_{1k}\exp(\eta_{2k}^2) \\ 0.3\eta_{2k}^3 \end{bmatrix}\right). \tag{59}$$

The performance index function is defined as in (6), where $Q = R = I \in \Re^{2\times 2}$ and I denotes the identity matrix.

We use neural networks to implement the generalized value iteration ADP algorithm. The structures of the critic and the action networks are chosen as 2–8–1 and 2–8–2, respectively. We choose a random array of state variable in $[-1, 1]$ to train the neural networks. For each iterative step, the critic network and the action network are trained for 2000 steps under the learning rate $\alpha = 0.005$ so that the approximation error limit 10^{-6} is reached. The generalized value iteration algorithm runs for 30 iterations to guarantee the convergence of the iterative performance index function. To illustrate the effectiveness of the algorithm, four different initial performance index functions are considered. Let the initial performance index function be the quadratic form which are expressed by $\Psi^j(z_k) = z_k^T P_j z_k$, $j = 1, \ldots, 4$. Let $P_1 = 0$. Let P_2–P_4 be initialized by positive definite matrices with the forms $P_2 = [9.07, -0.26; -0.26, 11.62]$, $P_3 = [10.48, 2.16; 2.16, 13.24]$, and $P_4 = [\,11.59, 0.61; 0.61, 13.40\,]$, respectively.

According to Theorem 4, we know that for arbitrary positive semi-definite function, the iterative performance index function will converge to the optimum. The curve of the iterative performance index functions under the four different initial performance index functions $\Psi^j(z_k)$, $j = 1, \ldots, 4$, are displayed in Fig. 2, which justify the convergence property of our developed algorithm.

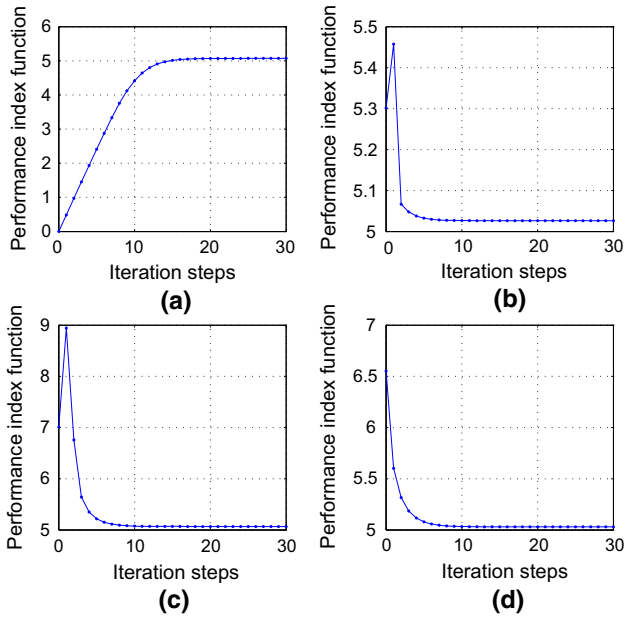

Fig. 2 The trajectories of the iterative performance index functions with $\Psi^j(z_k)$, $j = 1, 2, 3, 4$. **a** $\Psi^1(z_k)$. **b** $\Psi^2(z_k)$. **c** $\Psi^3(z_k)$. **d** $\Psi^4(z_k)$

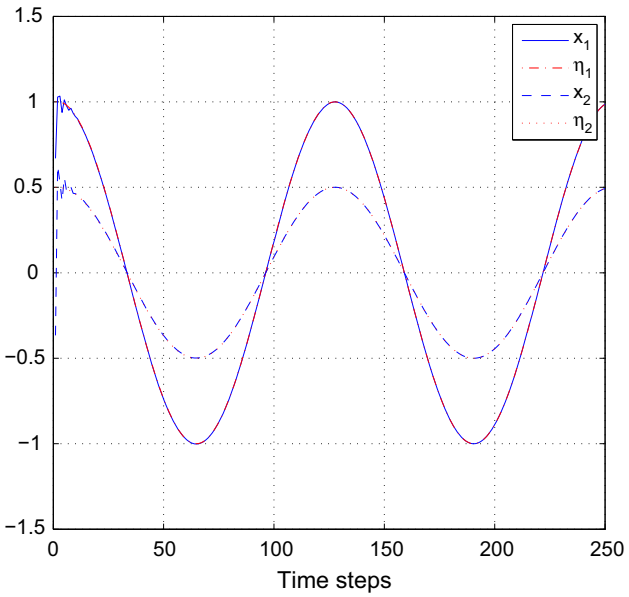

Fig. 3 The state and desired state trajectories

The optimal state and desired trajectories are shown in Fig. 3. The tracking error trajectories are shown in Fig. 4. The optimal control trajectories are shown in Fig. 5.

Remark 5 For $P_1 = 0$, we have $V_0^1(z_k) \equiv 0$. The generalized value iteration algorithm is then reduced to the traditional value iteration algorithm in Zhang et al. (2008). In Theorem 2 in Zhang et al. (2008), it shows that the iterative performance index function will be monotonically non-decreasing and converge to the optimum. From the simulation results we can see that the convergence properties of the traditional

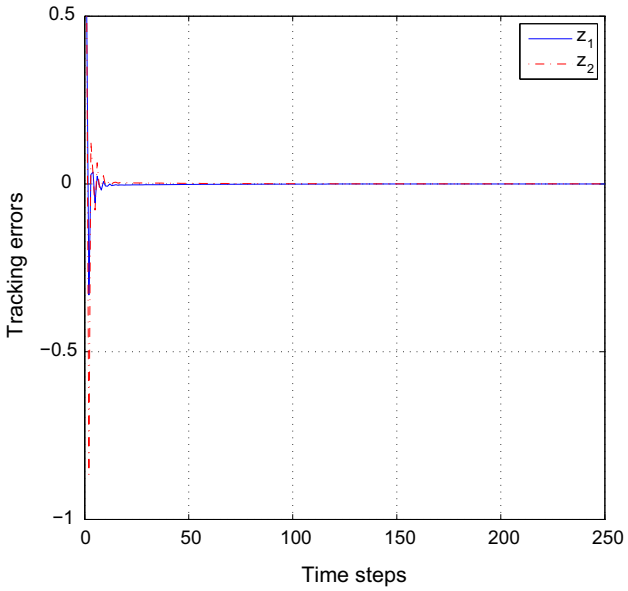

Fig. 4 The tracking error

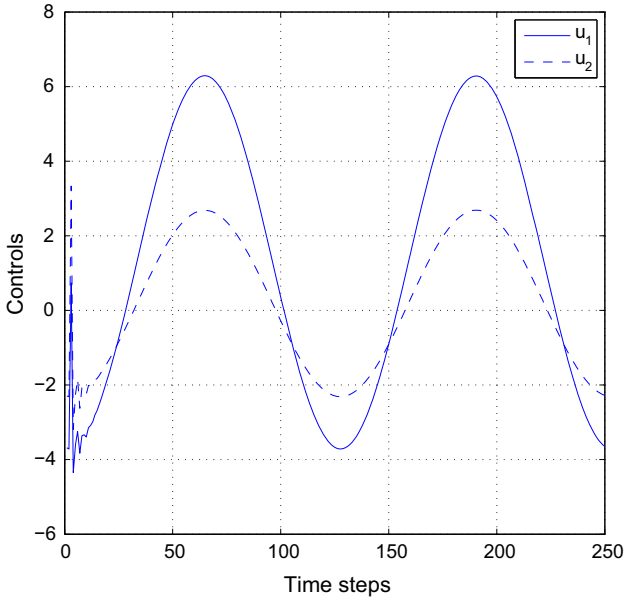

Fig. 5 The trajectories of the optimal controls

value iteration algorithm can be verified by the generalized value iteration algorithm.

6 Conclusion

In this paper, an effective generalized value iteration ADP algorithm is investigated to find the infinite horizon optimal tracking control law for a class of discrete-time nonlinear systems. In the developed iterative ADP algorithm, the initial performance index function can be chosen as an arbitrarily positive semi-definite function. Convergence property is developed to guarantee that the iterative performance index function will converge to the optimum. Neural networks are used to implement the proposed ADP algorithm. Finally, a simulation example is given to illustrate the performance of the developed algorithm.

Acknowledgments This work was supported in part by the National Natural Science Foundation of China under Grants 61034002, 61374105, 61233001, and 61273140, in part by Beijing Natural Science Foundation under Grant 4132078, in part by Fundamental Research Funds for the Central Universities under Grant FRF-TP-14-119A2, and in part by the Open Research Project from SKLMCCS under Grant 20120106.

References

Abu-Khalaf M, Lewis FL (2005) Nearly optimal control laws for nonlinear systems with saturating actuators using a neural network HJB approach. Automatica 41(5):779–791

Al-Tamimi A, Abu-Khalaf M, Lewis FL (2007) Adaptive critic designs for discrete-time zero-sum games with application to H_∞ control. IEEE Trans Syst Cybern Part B: Cybern 37(1):240–247

Al-Tamimi A, Lewis FL, Abu-Khalaf M (2008) Discrete-time nonlinear HJB solution using approximate dynamic programming: convergence proof. IEEE Trans Syst Man Cybern Part B: Cybern 38(4):943–949

Bhasin S, Kamalapurkar R, Johnson M, Vamvoudakis KG, Lewis FL, Dixon WE (2013) A novel actor-critic-identifier architecture for approximate optimal control of uncertain nonlinear systems. Automatica 49(1):82–92

Bellman RE (1957) Dynamic programming. Princeton University Press, Princeton

Bertsekas DP, Tsitsiklis JN (1996) Neuro-dynamic programming. Athena Scientific, Belmont

Bertsekas DP (2007) Dynamic programming and optimal control, 3rd edn. Athena Scientific, Belmont

Biswas S, Das S, Kundu S, Patra GR (2014) Utilizing time-linkage property in DOPs: an information sharing based artificial bee colony algorithm for tracking multiple optima in uncertain environments. Soft Comput 18(6):1199–1212

Chang HS (2013) On functional equations for Kth best policies in Markov decision processes. Automatica 49(1):297–300

Enns R, Si J (2003) Helicopter trimming and tracking control using direct neural dynamic programming. IEEE Trans Neural Netw 14(8):929–939

Fortier N, Sheppard J, Strasser S (2014) Abductive inference in Bayesian networks using distributed overlapping swarm intelligence. Soft Comput (in press). doi:10.1007/s00500-014-1310-0

Heydari A, Balakrishnan SN (2013) Finite-horizon control-constrained nonlinear optimal control using single network adaptive critics. IEEE Trans Neural Netw Learn Syst 24(1):145–157

Kouramas KI, Panos C, Faisca NP, Pistikopoulos EN (2013) An algorithm for robust explicit/multi-parametric model predictive control. Automatica 49(2):381–389

Kundu S, Das S, Vasilakos AV, Biswas S (2014) A modified differential evolution-based combined routing and sleep scheduling scheme for lifetime maximization of wireless sensor networks. Soft Comput (in press). doi:10.1007/s00500-014-1286-9

Lewis FL, Vrabie D, Vamvoudakis KG (2012) Reinforcement learning and feedback control: using natural decision methods to design optimal adaptive controllers. IEEE Control Syst 32(6):76–105

Lincoln B, Rantzer A (2006) Relaxing dynamic programming. IEEE Trans Autom Control 51(8):1249–1260

Liu D, Javaherian H, Kovalenko O, Huang T (2008) Adaptive critic learning techniques for engine torque and air-fuel ratio control. IEEE Trans Syst Man Cybern Part B Cybern 38(4):988–993

Liu D, Wei Q (2013) Finite-approximation-error-based optimal control approach for discrete-time nonlinear systems. IEEE Trans Cybern 43(2):779–789

Liu D, Wei Q (2014a) Multi-person zero-sum differential games for a class of uncertain nonlinear systems. Int J Adaptive Control Signal Process 28(3–5):205–231

Liu D, Wei Q (2014b) Policy iteration adaptive dynamic programming algorithm for discrete-time nonlinear systems. IEEE Trans Neural Netw Learn Syst 25(3):621–634

Liu D, Zhang Y, Zhang H (2005) A self-learning call admission control scheme for CDMA cellular networks. IEEE Trans Neural Netw 16(5):1219–1228

Mohler RR, Kolodziej WJ (1981) Optimal control of a class of nonlinear stochastic systems. IEEE Trans Autom Control 26(5):1048–1054

Murray JJ, Cox CJ, Lendaris GG, Saeks R (2002) Adaptive dynamic programming. IEEE Trans Syst Man Cybern Part C Appl Rev 32(2):140–153

Ni Z, He H (2013) Heuristic dynamic programming with internal goal representation. Soft Comput 17(11):2101–2108

Powell WB (2007) Approximate dynamic programming. Wiley, Hoboken

Prokhorov DV, Wunsch DC (1997) Adaptive critic designs. IEEE Trans Neural Netw 8(5):997–1007

Rubio JDJ (2014) Adaptive least square control in discrete time of robotic arms. Soft Comput (in press). doi:10.1007/s00500-014-1300-2

Rugh WJ (1971) System equivalence in a class of nonlinear optimal control problems. IEEE Trans Autom Control 16(2):189–194

Si J, Wang YT (2001) On-line learning control by association and reinforcement. IEEE Trans Neural Netw 12(2):264–276

Song R, Xiao W, Wei Q (2013) Multi-objective optimal control for a class of nonlinear time-delay systems via adaptive dynamic programming. Soft Comput 17(11):2109–2115

Song R, Xiao W, Wei Q, Sun C (2014) Neural-network-based approach to finite-time optimal control for a class of unknown nonlinear systems. Soft Comput 18(8):1645–1653

Sutton RS, Barto AG (1998) Reinforcement learning: an introduction. MIT Press, Cambridge

Wang F, Zhang H, Liu D (2009) Adaptive dynamic programming: an introduction. IEEE Comput Intell Mag 4(2):39–47

Wang F, Jin N, Liu D, Wei Q (2011) Adaptive dynamic programming for finite-horizon optimal control of discrete-time nonlinear systems with ϵ-error bound. IEEE Trans Neural Netw 22(1):24–36

Wei Q, Liu D (2012) An iterative ϵ-optimal control scheme for a class of discrete-time nonlinear systems with unfixed initial state. Neural Netw 32:236–244

Wei Q, Liu D (2013) Numerical adaptive learning control scheme for discrete-time nonlinear systems. IET Control Theory Appl 7(11):1472–1486

Wei Q, Wang D, Zhang D (2013) Dual iterative adaptive dynamic programming for a class of discrete-time nonlinear systems with time-delays. Neural Comput Appl 23(7–8):1851–1863

Wei Q, Liu D (2014a) Adaptive dynamic programming for optimal tracking control of unknown nonlinear systems with application to coal gasification. IEEE Trans Autom Sci Eng 11(4):1020–1036

Wei Q, Liu D (2014b) A novel iterative θ-adaptive dynamic programming for discrete-time nonlinear systems. IEEE Trans Autom Sci Eng 11(4):1176–1190

Wei Q, Liu D (2014c) Data-driven neuro-optimal temperature control of water gas shift reaction using stable iterative adaptive dynamic programming. IEEE Trans Ind Electron 61(11):6399–6408

Wei Q, Liu D (2014d) Stable iterative adaptive dynamic programming algorithm with approximation errors for discrete-time nonlinear systems. Neural Comput Appl 24(6):1355–1367

Wei Q, Liu D, Shi G (2014) A novel dual iterative Q-learning method for optimal battery management in smart residential environments. IEEE Trans Ind Electron (in press). doi:10.1109/TIE.2014.2361485

Wei Q, Wang F, Liu D, Yang X (2014) Finite-approximation-error based discrete-time iterative adaptive dynamic programming. IEEE Trans Cybern (in press). doi:10.1109/TCYB.2014.2354377

Wei Q, Zhang H, Dai J (2009) Model-free multiobjective approximate dynamic programming for discrete-time nonlinear systems with general performance index functions. Neurocomputing 72(7–9):1839–1848

Werbos PJ (1977) Advanced forecasting methods for global crisis warning and models of intelligence. General Syst Yearb 22:25–38

Werbos PJ (1991) A menu of designs for reinforcement learning over time. In: Miller WT, Sutton RS, Werbos PJ (eds) Neural Netw Control. MIT Press, Cambridge

Werbos PJ (1992) Approximate dynamic programming for real-time control and neural modeling. In: White DA, Sofge DA (eds) Handbook of intelligent control: neural, fuzzy, and adaptive approaches. Van Nostrand Reinhold, New York

Xu H, Jagannathan S (2013) Stochastic optimal controller design for uncertain nonlinear networked control system via neuro dynamic programming. IEEE Trans Neural Netw Learn Syst 24(3):471–484

Zhang H, Cui L, Luo Y (2013) Near-optimal control for nonzero-sum differential games of continuous-time nonlinear systems using single-network ADP. IEEE Trans Cybern 43(1):206–216

Zhang D, Liu D, Wang D (2014) Approximate optimal solution of the DTHJB equation for a class of nonlinear affine systems with unknown dead-zone constraints. Soft Comput 18(2):349–357

Zhang H, Luo Y, Liu D (2009) The RBF neural network-based near-optimal control for a class of discrete-time affine nonlinear systems with control constraint. IEEE Trans Neural Netw 20(9):1490–1503

Zhang H, Wei Q, Luo Y (2008) 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 Part B Cybern 38(4):937–942

Zhang H, Wei Q, Liu D (2011) An iterative adaptive dynamic programming method for solving a class of nonlinear zero-sum differential games. Automatica 47(1):207–214