www.ietdl.org

Published in IET Control Theory and Applications
Received on 4th September 2012
Revised on 24th January 2013
Accepted on 11th April 2013
doi: 10.1049/iet-cta.2012.0486

ISSN 1751-8644

Numerical adaptive learning control scheme for discrete-time non-linear systems

Qinglai Wei *Derong Liu*

The State Key Laboratory of Management and Control for Complex Systems, Institute of Automation, Chinese Academy of Sciences, Beijing 100190, People's Republic of China
E-mail: derong.liu@ia.ac.cn

Abstract: In this study, a novel numerical adaptive learning control scheme based on adaptive dynamic programming (ADP) algorithm is developed to solve numerical optimal control problems for infinite horizon discrete-time non-linear systems. Using the numerical controller, the domain of definition is constrained to a discrete set that makes the approximation errors always exist between the numerical controls and the accurate ones. Convergence analysis of the numerical iterative ADP algorithm is developed to show that the numerical iterative controls can make the iterative performance index functions converge to the greatest lower bound of all performance indices within a finite error bound under some mild assumptions. The stability properties of the system under the numerical iterative controls are proved, which allow the present iterative ADP algorithm to be implemented both on-line and off-line. Finally, two simulation examples are given to illustrate the performance of the present method.

1 Introduction

Optimal control of non-linear systems has always been the key focus of the control field in the latest several decades [1–4]. For the traditional optimal control schemes, in order to find the solution of optimal control problems, it often requires the control domain of definition to be a continuous space, such as the whole domain of real space [5]. This allows one to obtain optimal control solutions with arbitrary precision. As the development of digital computers, numerical control (NC) attracts more and more attention to researchers [6–8]. The NC concept employs a digital processing and output device, such as a digital computer, in control systems. Many functions of the conventional controller are replaced in an NC system by a computer program denoted as the NC program. In this situation, the accurate optimal control can hardly be obtained and there always exists an approximation error between the NC and the accurate one. This makes invalid the traditional optimal control schemes implemented by numerical controllers. To overcome this difficulty, a new optimal control method for NC systems must be considered.

Adaptive dynamic programming, proposed by Werbos [9, 10], is an effective adaptive learning control approach to solve optimal control problems for non-linear systems forward-in-time. Iterative adaptive dynamic programming (ADP) is a very important component of ADP to obtain the solution of Hamilton–Jacobi–Bellman (HJB) equation indirectly and have received lots of attention [11–17]. There are two main iterative ADP algorithms that are based on policy and value iterations, respectively [18]. Policy iteration algorithms for optimal control of continuous-time systems with continuous state and action spaces were given in [16]. In 2011, Wang *et al.* [19] studied finite-horizon optimal control problems of discrete-time non-linear systems with unspecified terminal time. Value iteration algorithms for optimal control of discrete-time non-linear systems were given in [20]. In [21], a value iteration algorithm, which is referred to as greedy iterative HDP algorithm, is proposed for finding the optimal control law and the convergence of the algorithm is also proved. While for value iteration algorithms, the stability of system under the iterative control cannot be guaranteed. In [22], an iterative θ-ADP algorithm is proposed, which permits the ADP algorithm to be implemented without the initial admissible control sequence, where the stability and convergence properties are both guaranteed. Although iterative ADP algorithms attract more and more attentions [23–31], for nearly all of the iterative ADP algorithms, the iterative control in each iteration is required to be accurately obtained. These iterative ADP algorithms can be called 'accurate iterative ADP algorithms'.

For the NC problem, however, the definition domain of the control (control set for brief) is always discrete, which means the accurate control cannot generally be obtained. In this situation, the convergence properties in the accurate iterative ADP algorithms may be invalid. Although in [32, 33], an iterative ADP algorithm with finite approximation error is proposed based on the iterative θ-ADP algorithm, while the domain of control is still required to be continuous. This means that the target control can always be defined. While for the NC problem, the target control may not belong to the discrete control set, which means that the target control may not be defined. Furthermore, for the algorithms in [32, 33] and many other iterative ADP algorithms, such as value iteration algorithms [20, 21, 27, 28], only the convergence of the iterative performance index function is discussed. In this situation, the stability of the iterative control cannot be guaranteed. This makes the iterative ADP algorithm in [20, 21, 27, 28, 32, 33] only implementable off-line. Till now, to the best of our knowledge, there are no discussions on the convergence and the stability properties of the iterative ADP algorithms implemented by a numerical controller. This presents a great challenge for on-line applications of iterative ADP algorithms, which motivates our research.

In this paper, it is the first time that the iterative ADP algorithm is studied with a numerical controller. Inspired by [22, 32, 33], a new numerical iterative θ-ADP algorithm is developed to obtain the numerical optimal control scheme by introducing a digital computer into the ADP structure. Convergence and optimality proofs are given to guarantee that the iterative performance index function converges to a finite neighbourhood of the optimal performance index function. We emphasise that the stability properties of iterative controls are also analysed, which make the non-linear system stable during the iteration of the numerical θ-iterative ADP algorithm. In summary, the main contributions of the present algorithm include:

1. A new numerical iterative ADP algorithm is developed to obtain the numerical optimal control law and optimal performance index function iteratively.
2. Convergence and optimality properties are proved to make the iterative performance index function converge to a finite neighbourhood of the optimal performance index function.
3. Any of the iterative controls obtained by the present numerical iterative ADP algorithm can stabilise the non-linear system. This makes the present algorithm implementable both on-line and off-line.
4. The least upper bound of the approximation error (admissible approximation error for brief) can also be obtained in order to justify quantitatively the convergence and stability properties.

2 Problem statement

In this paper, we study deterministic discrete-time non-linear systems

$$x_{k+1} = F(x_k, u_k), \quad k = 0, 1, 2, \ldots \tag{1}$$

where $x_k \in \mathbb{R}^n$ is the n-dimensional state vector and $u_k \in \mathbb{R}^m$ is the m-dimensional control vector. Let $\mathfrak{A} \subset \mathbb{R}^m$ be the set of NCs where we assume $0 \in \mathfrak{A}$. Let x_0 be the initial state. Let $\underline{u}_k = (u_k, u_{k+1}, \ldots)$ be an arbitrary control sequence from k to ∞. The performance index function for state x_0 under the control sequence $\underline{u}_0 = (u_0, u_1, \ldots)$ is defined as

$$J(x_0, \underline{u}_0) = \sum_{k=0}^{\infty} U(x_k, u_k) \tag{2}$$

where $U(x_k, u_k) > 0$ for $\forall x_k, u_k \neq 0$, is the utility function. In this paper, the results are based on the following assumptions.

Assumption 1: System (1) is controllable and the function $F(x_k, u_k)$ is Lipschitz continuous for $\forall x_k, u_k$.

Assumption 2: The system state $x_k = 0$ is an equilibrium state of system (1) under the control $u_k = 0$, i.e. $F(0, 0) = 0$.

Assumption 3: The feedback control $u_k = u(x_k)$ is a Lipschitz continuous function for $\forall x_k$ and satisfies $u_k = u(x_k) = 0$ for $x_k = 0$.

Assumption 4: The utility function $U(x_k, u_k)$ is a continuous positive-definite function for $\forall x_k, u_k$.

Define the set of control sequences as $\underline{\mathfrak{U}}_k = \{\underline{u}_k : \underline{u}_k = (u_k, u_{k+1}, \ldots), \forall u_{k+i} \in \mathbb{R}^m, i = 0, 1, \ldots\}$. Then, for an arbitrary control sequence $\underline{u}_k \in \underline{\mathfrak{U}}_k$, the optimal performance index function can be defined as

$$J^*(x_k) = \min\left\{J(x_k, \underline{u}_k) : \underline{u}_k \in \underline{\mathfrak{U}}_k\right\} \tag{3}$$

According to Bellman's principle of optimality, $J^*(x_k)$ satisfies the discrete-time HJB equation

$$J^*(x_k) = \min_{u_k \in \mathbb{R}^m}\{U(x_k, u_k) + J^*(F(x_k, u_k))\} \tag{4}$$

Then, the law of optimal single control vector can be expressed as

$$u^*(x_k) = \arg\min_{u_k \in \mathbb{R}^m}\{U(x_k, u_k) + J^*(F(x_k, u_k))\} \tag{5}$$

Generally, $J^*(x_k)$ is impossible to obtain by solving the HJB equation (4) directly. In [22], an iterative θ-ADP algorithm is proposed to solve the performance index function and control law by iterations. The iterative θ-ADP algorithm can be expressed as the following equations, with the iteration index i increasing from 0 to infinity.

$$\begin{cases} v_i(x_k) = \arg\min_{u_k \in \mathbb{R}^m}\{U(x_k, u_k) + V_i(x_{k+1})\} \\ V_{i+1}(x_k) = \min_{u_k \in \mathbb{R}^m}\{U(x_k, u_k) + V_i(x_{k+1})\} \end{cases} \tag{6}$$

where $V_0(x_k) = \theta\Psi(x_k)$, $\Psi(x_k) \in \bar{\Psi}_{x_k}$, is the initial performance index function and $\theta > 0$ is a large finite positive constant. The set of positive-definite functions $\bar{\Psi}_{x_k}$ is defined as follows [22, 32].

Definition 1: The set of initial positive-definite functions is defined as

$$\begin{aligned} \bar{\Psi}_{x_k} = \{&\Psi(x_k) : \Psi(x_k) > 0 \text{ is positive definite and} \\ &\exists \nu(x_k) \in \mathbb{R}^m, \Psi(F(x_k, \nu(x_k))) < \Psi(x_k)\}. \end{aligned} \tag{7}$$

In [22], it is proved that the iterative performance index function $V_i(x_k)$ converges to $J^*(x_k)$, as $i \to \infty$ (Theorem 3.5 in [22] for details). It also shows that for $i = 0, 1, \ldots$, the iterative control law $v_i(x_k)$ stabilises the non-linear system (1) when the accurate iterative control laws and accurate iterative performance index functions are obtained.

In real-world implementations, especially for the NC systems, however, for $\forall i = 0, 1, \ldots$, the accurate iterative control law $v_i(x_k)$ and the iterative performance index function $V_i(x_k)$ are generally impossible to obtain. For the situation that the control $u_k \in \mathfrak{A}$, the iterative θ-ADP algorithm in [22] may be invalid. First, for NC systems, the set of NCs $\mathfrak{A}$ is discrete. This means that there are always finite elements in the set of NCs $\mathfrak{A}$, which makes the iterative control law and iterative performance index function obtainable only with errors. Second, as $u_k \in \mathfrak{A}$, the convergence property of the iterative performance index function cannot be guaranteed, and the stability of the system under the control also cannot be proved. Furthermore, even if we solve the iterative control law and the iterative performance index function at every iteration step, it is not clear whether the error between the iterative performance index functions and $J^*(x_k)$ is finite for $\forall i$. These make the optimal performance index function and optimal control law nearly impossible to obtain by the iterative θ-ADP algorithm in [22].

3 Properties of the numerical iterative θ-ADP algorithm

In this section, a new numerical iterative θ-ADP algorithm is developed to obtain the numerical optimal controller for non-linear systems (1). Convergence and stability proofs will also be given in this section.

3.1 Derivation of the numerical iterative θ-ADP algorithm

In the present numerical iterative θ-ADP algorithm, the performance index functions and control laws are updated by iterations, with the iteration index i increasing from 0 to infinity. Then, for $\forall x_k \in \mathbb{R}^n$, let the initial performance index function be $\hat{V}_0(x_k) = \theta\Psi(x_k)$, where $\theta > 0$ is a large finite positive constant. The numerical iterative control law $\hat{v}_0(x_k)$ can be computed as follows

$$\hat{v}_0(x_k) = \arg\min_{u_k \in \mathfrak{A}}\{U(x_k, u_k) + \hat{V}_0(x_{k+1})\} \tag{8}$$

where $\hat{V}_0(x_{k+1}) = \theta\Psi(x_{k+1})$. The performance index function can be updated by

$$\begin{aligned}\hat{V}_1(x_k) &= \min_{u_k \in \mathfrak{A}}\{U(x_k, u_k) + \hat{V}_0(x_{k+1})\} \\ &= U(x_k, \hat{v}_0(x_k)) + \hat{V}_0(F(x_k, \hat{v}_0(x_k)))\end{aligned} \tag{9}$$

For $i = 1, 2, \ldots,$ the numerical iterative θ-ADP algorithm will iterate between the iterative control law

$$\hat{v}_i(x_k) = \arg\min_{u_k \in \mathfrak{A}}\{U(x_k, u_k) + \hat{V}_i(x_{k+1})\} \tag{10}$$

and the iterative performance index function

$$\begin{aligned}\hat{V}_{i+1}(x_k) &= \min_{u_k \in \mathfrak{A}}\{U(x_k, u_k) + \hat{V}_i(x_{k+1})\} \\ &= U(x_k, \hat{v}_i(x_k)) + \hat{V}_i(F(x_k, \hat{v}_i(x_k)))\end{aligned} \tag{11}$$

Remark 1: As the set of NCs $\mathfrak{A}$ is discrete, we can say that for $\forall i \geq 0$, $\hat{v}_i(x_k) \neq v_i(x_k)$ in general. Then, we have for $\forall i \geq 1$, the iterative performance index function $\hat{V}_i(x_k) \neq V_i(x_k)$, which means that there exists an error between $\hat{V}_i(x_k)$ and $V_i(x_k)$. It should be pointed out that the iterative approximation error is not a constant. The fact is that as the iteration index $i \to \infty$, the boundary of iterative approximation errors will also increase to infinity. The following lemma will show this property.

Lemma 1: Let $x_k \in \mathbb{R}^n$ be an arbitrary controllable state and Assumptions 1–4 hold. For $i = 1, 2, \ldots,$ define a new iterative performance index function as

$$\Gamma_i(x_k) = \min_{u_k \in \mathbb{R}^m}\{U(x_k, u_k) + \hat{V}_{i-1}(x_{k+1})\} \tag{12}$$

where $\hat{V}_i(x_k)$ is defined in (11). If the initial iterative performance index function $\Gamma_0(x_k) = \hat{V}_0(x_k) = \theta\Psi(x_k)$ and for $i = 1, 2, \ldots,$ there exists a finite constant ϵ that makes

$$\hat{V}_i(x_k) - \Gamma_i(x_k) \leq \epsilon \tag{13}$$

hold uniformly, then we have

$$\hat{V}_i(x_k) - V_i(x_k) \leq i\epsilon \tag{14}$$

where ϵ is called uniform approximation error.

Proof: The theorem can be proved by mathematical induction. First, let $i = 1$. We have

$$\begin{aligned}\Gamma_1(x_k) &= \min_{u_k \in \mathbb{R}^m}\{U(x_k, u_k) + \hat{V}_0(x_{k+1})\} \\ &= \min_{u_k \in \mathbb{R}^m}\{U(x_k, u_k) + V_0(F(x_k, u_k))\} \\ &= V_1(x_k)\end{aligned} \tag{15}$$

Then, according to (13), we can obtain

$$\hat{V}_1(x_k) - V_1(x_k) \leq \epsilon \tag{16}$$

Assume that (14) holds for $i - 1$. Then, for i, we have

$$\begin{aligned}\Gamma_i(x_k) &= \min_{u_k \in \mathbb{R}^m}\{U(x_k, u_k) + \hat{V}_{i-1}(x_{k+1})\} \\ &\leq \min_{u_k \in \mathbb{R}^m}\{U(x_k, u_k) + V_{i-1}(x_{k+1}) + (i-1)\epsilon\} \\ &= V_i(x_k) + (i-1)\epsilon\end{aligned} \tag{17}$$

Then, according to (13), we can obtain (14). □

Lemma 1 shows that although the approximation error for each single step is finite and may be small, as the iteration index $i \to \infty$ increases, the bounds of approximation errors between $\hat{V}_i(x_k)$ and $V_i(x_k)$ may also increase to infinity. To overcome these difficulties, we must discuss the convergence and stability properties of the iterative ADP algorithm with finite approximation errors. For convenience of analysis, we transform the expressions of the approximation error. According to the definitions of $\hat{V}_i(x_k)$ and $\Gamma_i(x_k)$ in (11) and (12), we have $\Gamma_i(x_k) \leq \hat{V}_i(x_k)$. Then, for $\forall i = 0, 1, \ldots,$ there exists a $\sigma \geq 1$ that makes

$$\Gamma_i(x_k) \leq \hat{V}_i(x_k) \leq \sigma\Gamma_i(x_k) \tag{18}$$

hold uniformly. Hence, we can give the following theorem.

Theorem 1: Let $x_k \in \mathbb{R}^n$ be an arbitrary controllable state and Assumptions 1–4 hold. For $\forall i = 0, 1, \ldots,$ let $\Gamma_i(x_k)$ be

doi: 10.1049/iet-cta.2012.0486

expressed as (12) and $\hat{V}_i(x_k)$ be expressed as (11). Let $\gamma < \infty$ and $1 \le \delta < \infty$ be constants that make

$$J^*(F(x_k, u_k)) \le \gamma U(x_k, u_k) \tag{19}$$

and

$$J^*(x_k) \le V_0(x_k) \le \delta J^*(x_k) \tag{20}$$

hold uniformly. If there exists a σ, that is, $1 \le \sigma < \infty$, that satisfies

$$\sigma \le 1 + \frac{\delta - 1}{\gamma\delta} \tag{21}$$

then the iterative performance index function $\hat{V}_i(x_k)$ converges to a bounded neighbourhood of $J^*(x_k)$, as $i \to \infty$.

Proof: According to (18)–(20), using the mathematical induction, for $\forall i = 0, 1, \ldots,$ the iterative performance index functions $\hat{V}_i(x_k)$ satisfies

$$\hat{V}_i(x_k) \le \sigma\left(1 + \sum_{j=1}^{i} \frac{\gamma^j \sigma^{j-1}(\sigma - 1)}{(\gamma + 1)^j} + \frac{\gamma^i \sigma^i (\delta - 1)}{(\gamma + 1)^i}\right) J^*(x_k) \tag{22}$$

From (22), we can see that for $j = 1, 2, \ldots,$ the sequence is a geometrical series. If (21) holds, then the right side of (22) is convergent for $i \to \infty$, which proves the conclusion. □

Theorem 2: Let $x_k \in \mathbb{R}^n$ be an arbitrary controllable state. If for $\forall x_k$, Theorem 1 holds and σ satisfies (21), then for $\forall i = 0, 1, \ldots,$ the numerical iterative control law $\hat{v}_i(x_k)$ is an asymptotically stable control law for system (1).

Proof: As $\hat{V}_0(x_k) = V_0(x_k) = \theta\Psi(x_k)$, we have that $\hat{V}_0(x_k)$ is a positive-definite function for $i = 0$. Using the mathematical induction, assume that the iterative performance index function $\hat{V}_i(x_k)$, $i = 0, 1, \ldots,$ is positive definite. Then, for $i + 1$, according to Assumptions 1–4, we can obtain

$$\hat{V}_{i+1}(0) = U(0, \hat{v}_i(0)) + \hat{V}_i(F(0, \hat{v}_i(0))) = 0 \tag{23}$$

for $x_k = 0$. When $x_k \to \infty$, as the utility function $U(x_k, u_k)$ is a positive function for x_k, we have $\hat{V}_{i+1}(x_k) \to \infty$. Hence, $\hat{V}_{i+1}(x_k)$ is a positive-definite function and the mathematical induction is completed. Next, let χ_i be defined as

$$\chi_i = \sigma\left(1 + \sum_{j=1}^{i} \frac{\gamma^j \sigma^{j-1}(\sigma - 1)}{(\gamma + 1)^j} + \frac{\gamma^i \sigma^i (\delta - 1)}{(\gamma + 1)^i}\right)$$

and $\bar{V}_i(x_k) = \chi_i J^*(x_k)$. As $\sigma \le 1 + (\delta - 1/\gamma\delta)$, we have $\chi_{i+1} \le \chi_i$, which means $\bar{V}_{i+1}(x_k) \le \bar{V}_i(x_k)$. Then, we can obtain

$$\bar{V}_i(x_k) \ge \bar{V}_{i+1}(x_k) = U(x_k, \bar{v}_i(x_k)) + \bar{V}_i(x_{k+1}) \tag{24}$$

where $\bar{v}_i(x_k) = \arg\min_{u_k \in \mathfrak{A}}\{U(x_k, u_k) + \bar{V}_i(x_{k+1})\}$. So, we have $\bar{V}_i(x_{k+1}) - \bar{V}_i(x_k) \le -U(x_k, \bar{v}_i(x_k)) < 0$. Hence, $\bar{V}_i(x_k)$ is a Lyapunov function and $\bar{v}_i(x_k)$ is an asymptotically stable control law for $\forall\, i = 0, 1, \ldots$.

As $\bar{v}_i(x_k)$ is an asymptotically stable control law, we have $x_{k+N} \to 0$ as $N \to \infty$, that is $\bar{V}_i(x_{k+N}) \to 0$. Since $0 < \hat{V}_i(x_k) \le \bar{V}_i(x_k)$ holds for $\forall x_k$, then we can obtain $0 < \hat{V}_i(x_{k+N}) \le \bar{V}_i(x_{k+N})$ as $N \to \infty$. So we have $\hat{V}_i(x_{k+N}) \to 0$ as $N \to \infty$. As $\hat{V}_i(x_k)$ is a positive-definite function, we can conclude $x_{k+N} \to 0$ as $N \to \infty$ under the control law $\hat{v}_i(x_k)$. □

3.2 Properties of the numerical iterative θ-ADP algorithm

Although Theorem 1 gives the convergence criterion, while we can see that the parameters σ, γ and δ are very difficult to achieve, which make the convergence criterion (21) quite difficult to verify. To overcome this difficulty, a new convergence conditions must be developed to guarantee the convergence of the numerical iterative θ-ADP algorithm. For the convenience of analysis, we define a new performance index function as

$$\hat{\mathcal{V}}_{i+1}(x_k, \hat{v}_i(x_k)) = U(x_k, \hat{v}_i(x_k)) + \hat{V}_i(F(x_k, \hat{v}_i(x_k))) \tag{25}$$

where we can see that $\hat{V}_{i+1}(x_k) = \hat{\mathcal{V}}_{i+1}(x_k, \hat{v}_i(x_k))$ and $\hat{V}_0(x_k) = \hat{\mathcal{V}}_0(x_k, 0)$. Then, we have the following definition.

Definition 2: The iterative performance index function $\hat{\mathcal{V}}_{i+1}(x_k, \hat{v}_i(x_k))$ is a Lipschitz continuous function for $\forall\, \hat{v}_i(x_k)$, if there exists an $L \in \mathbb{R}$ that makes

$$\|\hat{\mathcal{V}}_{i+1}(x_k, \hat{v}_i(x_k)) - \hat{\mathcal{V}}_{i+1}(x_k, \hat{v}'_i(x_k))\| \le L\|\hat{v}_i(x_k) - \hat{v}'_i(x_k)\| \tag{26}$$

hold, where $\hat{v}'_i(x_k) \in \mathfrak{A}$ and $\hat{v}'_i(x_k) \ne \hat{v}_i(x_k)$.

For the NC system, the set of NCs $\mathfrak{A}$ is discrete. Then, according to the grid principle, let P_j, $j = 1, 2, \ldots, m$, be the discrete grids for the jth dimension in $\mathfrak{A}$. Let Z be the set of positive integers. As $\mathfrak{A} \subset \mathbb{R}^m$, then using the grid principle, we can define $\mathfrak{A}$ as

$$\mathfrak{A} = \{u(p_1, \ldots, p_m) : p_1, \ldots, p_m \in Z, \quad 1 \le p_1 \le P_1, \ldots, 1 \le p_m \le P_m\} \tag{27}$$

to denote all the control elements in $\mathfrak{A}$, where $P_1, \ldots, P_m$ are all positive integers. Thus, for $\forall x_k \in \mathbb{R}^n$ and $\forall\, i = 0, 1, \ldots,$ there exists a sequence of positive numbers $p_1^i, \ldots, p_m^i$ that makes

$$u(p_1^i, \ldots, p_m^i) = \hat{v}_i(x_k) \tag{28}$$

hold, where $1 \le p_1^i \le P_1, \ldots, 1 \le p_m^i \le P_m$. Then, according to (25), we can rewrite $\hat{\mathcal{V}}_i(x_k, \hat{v}_i(x_k))$ as

$$\begin{aligned}\hat{\mathcal{V}}_{i+1}(x_k, \hat{v}_i(x_k)) &= \min_{u_k \in \mathfrak{A}}\{U(x_k, u_k) + \hat{V}_i(x_{k+1})\} \\ &= U(x_k, u(p_1^i, \ldots, p_m^i)) \\ &\quad + \hat{V}_i(F(x_k, u(p_1^i, \ldots, p_m^i)) \\ &= \hat{\mathcal{V}}_{i+1}(x_k, u(p_1^i, \ldots, p_m^i))\end{aligned} \tag{29}$$

Next, for $\forall u(p_1^i, \ldots, p_m^i) \in \mathfrak{A}$, $1 \le p_j^i \le P_j$, $j = 1, \ldots, m$, we can define a neighbourhood set of $u(p_1^i, \ldots, p_m^i)$ as

$$\bar{\mathfrak{A}}(p_1^i, \ldots, p_m^i) = \{u(\bar{p}_1^i, \ldots, \bar{p}_m^i) : j = 1, 2, \ldots, m, u(\bar{p}_1^i, \ldots, \bar{p}_m^i) \in \mathfrak{A}, |p_j^i - \bar{p}_j^i| \le \varrho\} \tag{30}$$

where $\varrho \in Z$ is defined as the radius of $\bar{\mathfrak{A}}(p_1^i, \ldots, p_m^i)$.

doi: 10.1049/iet-cta.2012.0486

Remark 2: From (30), we can see that for an m-dimensional control vector $u(p_1^i, \ldots, p_m^i)$ inside the set of NCs $\mathfrak{A}$ and for $\forall \varrho \geq 1$, there are $(2\varrho + 1)^m$ elements in the set $\bar{\mathfrak{A}}(p_1^i, \ldots, p_m^i)$, where $u(p_1^i, \ldots, p_m^i)$ is also included in $\bar{\mathfrak{A}}(p_1^i, \ldots, p_m^i)$. If $u(p_1^i, \ldots, p_m^i)$ is located at the boundary of $\mathfrak{A}$, the elements in $\bar{\mathfrak{A}}(p_1^i, \ldots, p_m^i)$ is reduced correspondingly. For the control vectors in the same neighbourhood set, for convenience of analysis, we assume that there exists the same Lipschitz constant. If the Lipschitz constants are different, the largest one will be adopted to guarantee the effectiveness of the algorithm and the details will be discussed later in the paper.

Define a new performance index function $\Upsilon_{i+1}(x_k, \tilde{v}_i(x_k))$ as

$$\begin{aligned}\Upsilon_{i+1}(x_k, \tilde{v}_i(x_k)) &= \min_{u_k \in \mathbb{R}^m} \{U(x_k, u_k) + \hat{V}_i(x_{k+1})\} \\ &= U(x_k, \tilde{v}_i(x_k)) + \hat{V}_i(F(x_k, \tilde{v}_i(x_k))) \end{aligned} \tag{31}$$

We can see that $\Gamma_{i+1}(x_k) = \Upsilon_{i+1}(x_k, \tilde{v}_i(x_k))$. As $\tilde{v}_i(x_k)$ cannot be obtained, it is very difficult to analyse its properties. While for $\forall\, u(p_1^i, \ldots, p_m^i)$, we can obtain $\bar{\mathfrak{A}}(p_1^i, \ldots, p_m^i)$ by (30) immediately. So, if $\tilde{v}_i(x_k)$ is inside $\bar{\mathfrak{A}}(p_1^i, \ldots, p_m^i)$ with the radius $\varrho \geq 1$, then the properties of $\tilde{v}_i(x_k)$ can be obtained. In the following, the relationship between $\tilde{v}_i(x_k)$ and $\bar{\mathfrak{A}}(p_1^i, \ldots, p_m^i)$ will be analysed. Before that, some lemmas are necessary.

Lemma 2: Let $O = (p_1^i, \ldots, p_m^i)$ denote the origin of the m-dimensional coordinate system and let $\overline{OL}$ be an arbitrary vector in the m-dimensional space. If we let ϑ_j, $j = 1, 2, \ldots, m$, be the intersection angle between $\overline{OL}$ and the jth coordinate axis, then we have $\sum_{j=1}^m \cos^2 \vartheta_j = 1$.

Proof: Let $L = (l_1^i, \ldots, l_m^i)$ be an arbitrary point in the m-dimensional coordinate system. Then we have $\overline{OL} = ((l_1^i - p_1^i), \ldots, (l_m^i - p_m^i))$. According to the definition of ϑ_j, we have $\cos \vartheta_j = (|l_j^i - p_j^i|)/(|\overline{OL}|)$, where

$$|\overline{OL}| = \sqrt{(l_1^i - p_1^i)^2 + \cdots + (l_m^i - p_m^i)^2}$$

Then we have

$$\sum_{j=1}^m \cos^2 \vartheta_j = \frac{(|l_1^i - p_1^i|)^2 + \cdots + (|l_m^i - p_m^i|)^2}{(l_1^i - p_1^i)^2 + \cdots + (l_m^i - p_m^i)^2} = 1$$

□

Lemma 3: Let $O = (p_1^i, \ldots, p_m^i)$ denote the origin of the m-dimensional coordinate system and let $\overline{OL}$ be an arbitrary vector in the m-dimensional space. Let $A_j, j = 1, 2 \ldots, m$, be points on the jth coordinate axis of m-dimensional space and $\forall j = 1, 2, \ldots, m$, $\overline{OA_j} = OL$. If we let $\vartheta_j = \min\{\vartheta_1, \ldots, \vartheta_m\}$, then we have

$$\overline{A_j L} \leq \sqrt{\frac{m-1}{m}}\overline{OL} \Bigg/ \cos\left(\frac{1}{2}\arcsin\left(\sqrt{\frac{m-1}{m}}\right)\right) \tag{32}$$

Proof: Let $\vartheta_1 = \cdots = \vartheta_m = \arccos(1/\sqrt{m})$, then we can see that ϑ_j reaches the maximum. Let $\alpha_j = \angle OA_jL$. We can obtain $\alpha_j = \frac{1}{2}(\pi - \vartheta_j)$. According to sine rule [34], we have

$$\overline{A_j L} \leq \frac{\sin \vartheta_j}{\sin \alpha_j}\overline{OL} \tag{33}$$

As $\cos \vartheta_j = (1/m)$, we have $\sin \vartheta_j = \sqrt{(m - 1/m)}$, and $\sin \alpha_j = \cos(1/2 \arcsin(\sqrt{(m - 1/m)}))$. Taking $\sin \vartheta_j$ and $\sin \alpha_j$ into (33), we can obtain the conclusion. □

Lemma 4: Let $O = (p_1^i, \ldots, p_m^i)$ denote the origin of the m-dimensional coordinate system and let $A_\ell = ({}_\ell p_1^i, \ldots, {}_\ell p_m^i)$ be an arbitrary point in $\bar{\mathfrak{A}}(p_1^i, \ldots, p_m^i)$, where $1 \leq \ell \leq (2\varrho + 1)^m$. Let $L = (\bar{p}_1^i, \ldots, \bar{p}_m^i)$ be an arbitrary point that satisfies $\overline{OL} \geq \max_{1 \leq \ell \leq (2\varrho+1)^m}\{\overline{OA}_\ell\}$. If the radius ϱ of $\bar{\mathfrak{A}}(p_1^i, \ldots, p_m^i)$ satisfies the following inequality

$$\varrho \geq \frac{3(m-1) + \sqrt{3(m-1)}}{3m} \tag{34}$$

then there exists an ℓ that satisfies

$$\overline{A_\ell L} \leq \overline{OL} \tag{35}$$

Proof: Without loss of generality, let $L = (\bar{p}_1^i, \ldots, \bar{p}_m^i)$ be located in the first quadrant. According to Lemmas 2 and 3, we can see that if intersection angles satisfy $\vartheta_1 = \cdots = \vartheta_m = \arccos(1/\sqrt{m})$, the value on the left-hand side of (32) reaches its maximum for each coordinate axis. In this situation, we can let $L = (p_1^i + \varrho, \ldots, p_m^i + \varrho)$. Let $A_\ell = (0, \ldots, 0, p_m^i + \varrho)$ and $A'_\ell = (p_1^i + \varrho - 1, \ldots, p_{m-1}^i + \varrho - 1, p_m^i + \varrho)$. Then, we can see that the points A_ℓ, A'_ℓ and L are on the same line. Let $\vartheta'_\ell = \angle B'_\ell OL$. Then, we have

$$\cos \vartheta'_\ell = \frac{(\overline{OA'_\ell})^2 + (\overline{OL})^2 - (\overline{LA'_\ell})^2}{2\overline{OA'_\ell}\,\overline{OL}} \tag{36}$$

Taking the coordinates of A'_ℓ and L into (36), we can obtain

$$\cos \vartheta'_\ell = \frac{(\varrho - 1)m + 1}{\sqrt{m}\sqrt{(\varrho - 1)^2 m - (\varrho - 1)^2 + \varrho^2}} \tag{37}$$

Next, extend the line $\overline{OA'_\ell}$ to B'_ℓ, so that it satisfies $\overline{OB'_\ell} = \overline{OL}$. Then $\triangle OB'_\ell L$ is an isosceles triangle. It is obvious that when $\vartheta'_\ell \leq \pi/3$, that is $\cos \vartheta'_\ell \geq 1/2$, we can obtain $\overline{B'_\ell L} \leq \overline{OL}$. According to (37), we can obtain (34). On the other hand, according to sine rule [31], we can obtain $(\sin \vartheta'_\ell)/(\overline{B'_\ell L}) = (\sin \angle OB'_\ell L)/(\overline{OL})$, and $(\sin \vartheta'_\ell)/(\overline{A'_\ell L}) = (\sin \angle OA'_\ell L)/(\overline{OL})$. If $\angle OB'_\ell L \leq \angle OA'_\ell L \leq \pi/2$, we can easily obtain $\overline{A'_\ell L} \leq \overline{B'_\ell L} \leq \overline{OL}$. If $\pi/2 \leq \angle OA'_\ell L \leq \pi$, then $\angle OA'_\ell L$ is the maximum angle in $\triangle OA'_\ell L$, which obtain $\overline{A'_\ell L} \leq \overline{OL}$ directly. The proof is completed. □

Given the above preparation, we derive the following theorem.

Theorem 3: Let $\hat{v}_i(x_k) = u(p_1^i, p_2^i, \ldots, p_m^i) \in \mathfrak{A}$ and let $u({}_\ell p_1^i, {}_\ell p_2^i, \ldots, {}_\ell p_m^i) \in \bar{\mathfrak{A}}(p_1^i, p_2^i, \ldots, p_m^i)$, $1 \leq \ell \leq (2\varrho + 1)^m$, be an arbitrary control vector. If the radius ϱ of $\bar{\mathfrak{A}}(p_1^i, p_2^i, \ldots, p_m^i)$ satisfies (34), then there exists a positive number $L \in \mathbb{R}$ that satisfies

$$\begin{aligned}&\|\hat{\mathcal{V}}_{i+1}(x_k, u(p_1^i, \ldots, p_m^i)) - \Upsilon_{i+1}(x_k, \tilde{v}_i(x_k))\| \\ &\quad \leq L \max_{1 \leq \ell \leq (2\varrho+1)^m}\{\|u(p_1^i, \ldots, p_m^i) - u({}_\ell p_1^i, \ldots, {}_\ell p_m^i)\|\} \end{aligned} \tag{38}$$

IET Control Theory Appl., 2013, Vol. 7, Iss. 11, pp. 1472–1486
doi: 10.1049/iet-cta.2012.0486

Proof: According to the definitions of the iterative performance index functions $\hat{\mathcal{V}}_{i+1}(x_k, u(p_1^i, \ldots, p_m^i))$ and $\Upsilon_{i+1}(x_k, \tilde{v}_i(x_k))$ in (29) and (31), respectively, we can see that if we put the control $\tilde{v}_i(x_k)$ into the set of NCs $\mathfrak{A}$, then we have $\hat{\mathcal{V}}_i(x_k, \tilde{v}_i(x_k)) = \Upsilon_i(x_k, \tilde{v}_i(x_k))$. For the control $u(p_1^i, \ldots, p_m^i)$, according to Definition 2, there exists an L that makes

$$\begin{aligned}\|\hat{\mathcal{V}}_{i+1}(x_k, u(p_1^i, \ldots, p_m^i)) - \Upsilon_{i+1}(x_k, \tilde{v}_i(x_k))\| \\ \leq L\|u(p_1^i, \ldots, p_m^i) - \tilde{v}_i(x_k)\|\end{aligned} \tag{39}$$

hold. As $\tilde{v}_i(x_k)$ cannot be obtained accurately, the distance between $\tilde{v}_i(x_k)$ and $u(p_1^i, \ldots, p_m^i)$ is unknown. Hence, $\tilde{v}_i(x_k)$ must be replaced by other known vector. Next, we will show that

$$\begin{aligned}&\|u(p_1^i, \ldots, p_m^i) - \tilde{v}_i(x_k)\| \\ &\quad \leq \max_{1\leq \ell \leq (2\varrho+1)^m}\{\|u(p_1^i, \ldots, p_m^i) - u({}_\ell p_1^i, \ldots, {}_\ell p_m^i)\|\}.\end{aligned} \tag{40}$$

As $\tilde{v}_i(x_k)$ is put into $\mathfrak{A}$, then $\tilde{v}_i(x_k)$ becomes the neighbouring point of $u(p_1^i, \ldots, p_m^i)$ and we can put $\tilde{v}_i(x_k)$ into $\bar{\mathfrak{A}}(p_1^i, \ldots, p_m^i)$ which makes $\tilde{v}_i(x_k) \in \bar{\mathfrak{A}}(p_1^i, \ldots, p_m^i)$. Next, we will prove the conclusion by contradiction. Assume that the inequality (40) does not hold. Then, we have

$$\begin{aligned}&\|u(p_1^i, \ldots, p_m^i) - \tilde{v}_i(x_k)\| \\ &\quad > \max_{1\leq \ell \leq (2\varrho+1)^m}\{\|u(p_1^i, \ldots, p_m^i) - u({}_\ell p_1^i, \ldots, {}_\ell p_m^i)\|\}\end{aligned} \tag{41}$$

as $\tilde{v}_i(x_k)$ belongs to the set $\bar{\mathfrak{A}}(p_1^i, \ldots, p_m^i)$. As there are m dimensions in $\bar{\mathfrak{A}}(p_1^i, \ldots, p_m^i)$, we can divide it into 2^m quadrants.

Without loss of generality, let $\tilde{v}_i(x_k)$ be located in the first quadrant where $L = (\tilde{p}_1^i, \ldots, \tilde{p}_m^i)$ is the corresponding coordinate. If we let $O = (p_1^i, \ldots, p_m^i)$ be the origin, then we have $\overline{OL} = \|u(p_1^i, \ldots, p_m^i) - \tilde{v}_i(x_k)\|$. As (34) holds, according to Theorem 4, if $\overline{OL}$ is the max vector in $\bar{\mathfrak{A}}(p_1^i, \ldots, p_m^i)$, then there exists an vector $OA'_\ell \in \bar{\mathfrak{A}}(p_1^i, \ldots, p_m^i)$ with the coordinate $A'_\ell = ({}_\ell p_1^i, \ldots, {}_\ell p_m^i)$, that makes (35) hold. Let L_1 be the Lipschitz constant. Then we can obtain

$$\begin{aligned}&\left\|\hat{\mathcal{V}}_{i+1}(x_k, u(p_1^i, \ldots, p_m^i)) - \Upsilon_{i+1}(x_k, \tilde{v}_i(x_k))\right\| \\ &\quad = L_1\|u(p_1^i, \ldots, p_m^i) - \tilde{v}_i(x_k)\| \\ &\quad \geq L_1\|u({}_\ell p_1^i, \ldots, {}_\ell p_m^i) - \tilde{v}_i(x_k)\| \\ &\quad = \left\|\hat{\mathcal{V}}_{i+1}(x_k, u_\ell^i({}_\ell p_1^i, \ldots, {}_\ell p_m^i)) - \Upsilon_{i+1}(x_k, \tilde{v}_i(x_k))\right\|\end{aligned} \tag{42}$$

According to the definition of $\Upsilon_{i+1}(x_k, \tilde{v}_i(x_k))$ in (31), we know that $\hat{\mathcal{V}}_{i+1}(x_k, u(p_1^i, \ldots, p_m^i)) \geq \Upsilon_{i+1}(x_k, \tilde{v}_i(x_k))$ and $\hat{\mathcal{V}}_{i+1}(x_k, u_\ell^i({}_\ell p_1^i, \ldots, {}_\ell p_m^i)) \geq \Upsilon_{i+1}(x_k, \tilde{v}_i(x_k))$ hold, for $\forall i = 0, 1, \ldots$. Thus, according to (42), we can obtain

$$\hat{\mathcal{V}}_{i+1}(x_k, u(p_1^i, \ldots, p_m^i)) > \hat{\mathcal{V}}_{i+1}(x_k, u_\ell^i({}_\ell p_1^i, \ldots, {}_\ell p_m^i)) \tag{43}$$

This contradicts with the definition of $\hat{\mathcal{V}}_{i+1}(x_k, u(p_1^i, \ldots, p_m^i))$ in (29). Therefore the assumption is false and the inequality (40) holds. The proof is completed. □

Remark 3: From Theorem 3 we can see that to obtain (38), the inequality (34) should be satisfied. While according to (30), for $\forall \varrho = 1, 2, \ldots$, the elements in the set $\bar{\mathfrak{A}}(p_1^i, \ldots, p_m^i)$ is exponentially increasing. This may cause serious computational burden. Fortunately, this situation will never happen. From (34), we can see that if the dimension $1 \leq m \leq 4$, we have $\varrho \leq 1$, which means only the points in $\bar{\mathfrak{A}}(p_1^i, \ldots, p_m^i)$ with radius $\varrho = 1$ will be calculated. For $4 < m < \infty$, we can see that $1 < (3(m-1) + \sqrt{3(m-1)}/3m) < 2$. This means for $m > 4$, the radius $\varrho = 2$ is enough. Therefore the number of the points that needs to compute in $\bar{\mathfrak{A}}(p_1^i, \ldots, p_m^i)$ is never more than 5^m, which releases the computation burden very much.

According to the definitions of the iterative performance index functions $\hat{\mathcal{V}}_{i+1}(x_k, u(p_1^i, \ldots, p_m^i))$ and $\Upsilon_{i+1}(x_k, \tilde{v}_i(x_k))$ in (29) and (31), respectively, for $\forall i = 0, 1, \ldots,$ we can define

$$\hat{\mathcal{V}}_{i+1}(x_k, u(p_1^i, \ldots, p_m^i)) - \Upsilon_{i+1}(x_k, \tilde{v}_i(x_k)) = \epsilon_{i+1}(x_k) \tag{44}$$

where $\epsilon_0(x_k) = 0$. Then, for any $\epsilon_{i+1}(x_k)$ expressed in (44), there exists a $\sigma_{i+1}(x_k)$ that satisfies

$$\begin{aligned}\Upsilon_{i+1}(x_k, \tilde{v}_i(x_k)) &= \hat{\mathcal{V}}_{i+1}(x_k, u(p_1^i, \ldots, p_m^i)) - \epsilon_{i+1}(x_k) \\ &= \frac{\hat{\mathcal{V}}_{i+1}(x_k, u(p_1^i, \ldots, p_m^i))}{\sigma_{i+1}(x_k)}\end{aligned} \tag{45}$$

Theorem 4: Let the iterative performance index function $\hat{\mathcal{V}}_{i+1}(x_k, u(p_1^i, \ldots, p_m^i))$ and the numerical iterative control $u(p_1^i, \ldots, p_m^i)$ be obtained by (28)–(29), respectively. If for $x_k \in \mathbb{R}^n$, we define the admissible approximation error as

$$\begin{aligned}\bar{\epsilon}_{i+1}(x_k) = L_i(p_1^i, \ldots, p_m^i) \max_{1\leq \ell \leq (2\varrho+1)^m} \\ \times \{\|u(p_1^i, \ldots, p_m^i) - u({}_\ell p_1^i, \ldots, {}_\ell p_m^i)\|\}\end{aligned} \tag{46}$$

where $L_i(p_1^i, \ldots, p_m^i)$ is Lipschitz constant and $\bar{\epsilon}_0(x_k) = 0$, then for $\forall\, i = 0, 1, \ldots,$ we have

$$\epsilon_{i+1}(x_k) \leq \bar{\epsilon}_{i+1}(x_k) \tag{47}$$

Proof: As $\hat{\mathcal{V}}_{i+1}(x_k, u(p_1^i, \ldots, p_m^i))$ is Lipschitz continuous, according to (27), we have

$$\begin{aligned}&\left\|\hat{\mathcal{V}}_{i+1}(x_k, u(p_1^i, \ldots, p_m^i)) - \Upsilon_{i+1}(x_k, \tilde{v}_i(x_k))\right\| \\ &\quad \leq L_i(p_1^i, \ldots, p_m^i)(u(p_1^i, \ldots, p_m^i) - \tilde{v}_i(x_k))\end{aligned} \tag{48}$$

where $L_i(p_1^i, \ldots, p_m^i)$ is the Lipschitz constant. According to (40), we can draw the conclusion. □

For the set of NC $\mathfrak{A}$ expressed in (27), as $\mathfrak{A}$ is known, then for any control $u(p_1^i, \ldots, p_m^i) \in \mathfrak{A}$, we can obtain $\bar{\mathfrak{A}}(p_1^i, \ldots, p_m^i)$ immediately. Hence, if the Lipschitz constant $L_i(p_1^i, \ldots, p_m^i)$ is known, then the error $\bar{\epsilon}_{i+1}$ can be obtained by (46). Next, we will give a method to obtain $L_i(p_1^i, \ldots, p_m^i)$. Let $u({}_\ell p_1^i, \ldots, {}_\ell p_m^i) \in \bar{\mathfrak{A}}(p_1^i, \ldots, p_m^i)$ be an arbitrary control vector. Then, we can obtain

$$\left\| \hat{\mathcal{V}}_{i+1}(x_k, u(p_1^i, \ldots, p_m^i)) - \hat{\mathcal{V}}_{i+1}(x_k, u({}_\ell p_1^i, \ldots, {}_\ell p_m^i)) \right\| = \bar{L}_i^\ell(p_1^i, \ldots, p_m^i) \| u(p_1^i, \ldots, p_m^i) - u({}_\ell p_1^i, \ldots, {}_\ell p_m^i) \| \quad (49)$$

where $\bar{L}_i^\ell(p_1^i, \ldots, p_m^i) > 0$, $\ell = 1, 2, \ldots, (2\varrho + 1)^m$, $i = 0, 1, \ldots$. Let

$$\bar{L}_i(p_1^i, \ldots, p_m^i) = \max_{1 \le \ell \le (2\varrho+1)^m} \{\bar{L}_i^\ell(p_1^i, \ldots, p_m^i)\} \quad (50)$$

be the local Lipschitz constant. Then (49) can be written as

$$\left\| \hat{\mathcal{V}}_{i+1}(x_k, u(p_1^i, \ldots, p_m^i)) - \hat{\mathcal{V}}_{i+1}(x_k, u({}_\ell p_1^i, \ldots, {}_\ell p_m^i)) \right\| \le \bar{L}_i(p_1^i, \ldots, p_m^i) \| u(p_1^i, \ldots, p_m^i) - u({}_\ell p_1^i, \ldots, {}_\ell p_m^i) \| \quad (51)$$

For $\forall\, u(p_1^i, \ldots, p_m^i) \in \mathfrak{A}$, we can define the global Lipschitz constant $\bar{L}_i$ as

$$\bar{L}_i = \max\{\bar{L}_i(p_1^i, \ldots, p_m^i) : 1 \le p_j^i \le P_j,\ j = 1, 2, \ldots, m\} \quad (52)$$

Thus, from (50) and (52), we can easily obtain

$$L_i(p_1^i, \ldots, p_m^i) \le \bar{L}_i \quad (53)$$

Remark 4: From (49)–(52), we can see that if we want to obtain the global Lipschitz constant $\bar{L}_i$, then all the controls $u(p_1^i, \ldots, p_m^i)$ in the set of NCs $\mathfrak{A}$ should be searched. This means for $\forall\, i = 0, 1, \ldots$, we should run $P_1 \cdot P_2 \cdot \cdots \cdot P_m \cdot (2\varrho + 1)^m$ times of computation to obtain $\bar{L}_i$ by (52). So, the computational burden is very heavy. In this paper, to simplify the algorithm, the local Lipschitz constant $L_i(p_1^i, \ldots, p_m^i)$ is used instead of the global one. We use the control vector $u(p_1^i, \ldots, p_m^i)$ which satisfies (28). Obtaining $\bar{\mathfrak{A}}(p_1^i, \ldots, p_m^i)$ according to (30), we can obtain $\bar{L}_i(p_1^i, \ldots, p_m^i)$ by (50). Then, the approximation error $\bar{\epsilon}_{i+1}(x_k)$ can easily be obtained by (46).

In the above, we give an effective method to obtain the approximation error $\bar{\epsilon}_{i+1}(x_k)$ of the numerical iterative θ-ADP algorithm. In the following, we will show how to obtain the admissible approximation error to guarantee the convergence criterion of the present numerical iterative ADP algorithm. According to (19), we can define $\gamma = \max\{J^*(F(x_k, u_k))/U(x_k, u_k) : x_k \in \mathbb{R}^n, u_k \in \mathfrak{A}\}$. If we let

$$\tilde{\gamma}_{i+1} = \left\{ \frac{V_i(F(x_k, u_k))}{U(x_k, u_k)} : x_k \in \mathbb{R}^n, u_k \in \mathfrak{A} \right\} \quad (54)$$

then we can obtain $\tilde{\gamma}_{i+1} \ge \gamma$. Before the next theorem, we make some denotations. Let

$$\bar{\sigma}_{i+1}(x_k) = \frac{\hat{\mathcal{V}}_{i+1}(x_k, u(p_1^i, \ldots, p_m^i))}{\hat{\mathcal{V}}_{i+1}(x_k, u(p_1^i, \ldots, p_m^i)) - \bar{\epsilon}_{i+1}(x_k)} \quad (55)$$

and

$$\delta_{i+1}(x_k) = \frac{\hat{\mathcal{V}}_0(x_k, 0)}{\hat{\mathcal{V}}_{i+1}(x_k, u(p_1^i, \ldots, p_m^i))} \quad (56)$$

Theorem 5: Let the iterative performance index function $\hat{\mathcal{V}}_{i+1}(x_k, u(p_1^i, \ldots, p_m^i))$ be defined in (25) and the numerical iterative control $u(p_1^i, \ldots, p_m^i)$ be defined in (28). For $\forall i = 0, 1, \ldots$, if the approximation error satisfies

$$\bar{e}_{i+1}(x_k) \le \frac{\mathcal{V}_{i+1}(x_k, u(p_1^i, \ldots, p_m^i))}{\hat{\mathcal{V}}_0(x_k, 0)(\tilde{\gamma}_{i+1} + 1) - \mathcal{V}_{i+1}(x_k, u(p_1^i, \ldots, p_m^i))} \times (\hat{\mathcal{V}}_0(x_k, 0) - \mathcal{V}_{i+1}(x_k, u(p_1^i, \ldots, p_m^i))) \quad (57)$$

then we have the numerical iterative control law $u(p_1^i, \ldots, p_m^i)$ stabilises the non-linear system (1) and simultaneously makes the iterative performance index function $\hat{\mathcal{V}}_{i+1}(x_k, u(p_1^i, \ldots, p_m^i))$ converge to a finite neighbourhood of $J^*(x_k)$, as $i \to \infty$.

Proof: From (20), we can define $\delta = \max\{(V_0(x_k)/J^*(x_k)) : x_k \in \mathbb{R}^n\}$. From (54)–(56), for $\forall\, x_k \in \mathbb{R}^n$, we can obtain

$$\begin{cases} \sigma_{i+1}(x_k) \le \bar{\sigma}_{i+1}(x_k) \\ 1 + \dfrac{\delta_{i+1}(x_k) - 1}{\tilde{\gamma}_{i+1}\delta_{i+1}(x_k)} \le 1 + \dfrac{\delta - 1}{\gamma\delta} \end{cases} \quad (58)$$

So, if

$$\bar{\sigma}_{i+1}(x_k) \le 1 + \frac{\delta_{i+1}(x_k) - 1}{\tilde{\gamma}_{i+1}\delta_{i+1}(x_k)} \quad (59)$$

holds, then we have (58) holds. Putting (55) and (56) into (59), we can obtain (57).

On the other hand, according to (54) and the definition of σ in (18), we have

$$\sigma = \max_{x_k \in \mathbb{R}^n, i=0,1,\ldots} \{\sigma_{i+1}(x_k)\} \le \max_{x_k \in \mathbb{R}^n, i=0,1,\ldots} \{\bar{\sigma}_{i+1}(x_k)\} \le \max_{x_k \in \mathbb{R}^n, i=0,1,\ldots} \left\{ 1 + \frac{\delta_{i+1}(x_k) - 1}{\tilde{\gamma}_{i+1}\delta_{i+1}(x_k)} \right\} \le 1 + \frac{\delta - 1}{\gamma\delta} \quad (60)$$

According to Theorems 1 and 2, we can draw the conclusion. □

Theorem 6: Let the iterative performance index function $\hat{\mathcal{V}}_{i+1}(x_k, u(p_1^i, \ldots, p_m^i))$ be defined in (25) and the numerical iterative control $u(p_1^i, \ldots, p_m^i)$ be defined in (28). If for $\forall\, x_k \in \mathbb{R}^n$, we have

$$U(x_k, u_k) \ge U(x_k, 0) \quad (61)$$

and for $\forall\, i = 0, 1, \ldots$, the approximation error satisfies (see (62))

then we have that the numerical iterative control law $u(p_1^i, \ldots, p_m^i)$ stabilises the non-linear system (1) and simultaneously makes the iterative performance index functions $\hat{\mathcal{V}}_{i+1}(x_k, u(p_1^i, \ldots, p_m^i))$ converge to a finite neighbourhood of $J^*(x_k)$, as $i \to \infty$.

$$\bar{e}_{i+1}(x_k) \le \frac{\mathcal{V}_{i+1}^2(x_k, u(p_1^i, \ldots, p_m^i))(\hat{\mathcal{V}}_0(x_k, 0) - \mathcal{V}_{i+1}(x_k, u(p_1^i, \ldots, p_m^i)))}{\hat{\mathcal{V}}_0(x_k, 0)(\mathcal{V}_{i+1}(x_k, u(p_1^i, \ldots, p_m^i)) - U(x_k, 0)) + \mathcal{V}_{i+1}(x_k, u(p_1^i, \ldots, p_m^i))(\hat{\mathcal{V}}_0(x_k, 0) - \mathcal{V}_{i+1}(x_k, u(p_1^i, \ldots, p_m^i)))} \quad (62)$$

IET Control Theory Appl., 2013, Vol. 7, Iss. 11, pp. 1472–1486
doi: 10.1049/iet-cta.2012.0486

Proof: If we let

$$\hat{\gamma}_{i+1} = \max\left\{\frac{\mathcal{V}_{i+1}(x_k, u(p_1^i, \ldots, p_m^i))}{U(x_k, 0)} - 1 : x_k \in \mathbb{R}^n, u_k \in \mathfrak{A}\right\} \tag{63}$$

then we can obtain $\hat{\gamma}_{i+1} \geq (J^*(x_k))/(U(x_k, u^*(x_k))) - 1 \geq \gamma$. So, if $\bar{\sigma}_{i+1}(x_k) \leq 1 + (\delta_{i+1}(x_k) - 1/\hat{\gamma}_{i+1}\delta_{i+1}(x_k))$, then we have (21) holds, which means that iterative performance index functions $\hat{\mathcal{V}}_{i+1}(x_k, u(p_1^i, \ldots, p_m^i))$ converge to a finite neighbourhood of $J^*(x_k)$ according to Theorem 1. According to (55), (56) and (63), and we can obtain (62). □

Remark 5: We can see that if the utility function $U(x_k, u_k)$ satisfies (61), then the convergence criterion in (62) becomes much simpler than the one in (57). We should say that many real utility functions satisfy the criterion in (62). For example, all the quadratic form utility functions (or state-depended ones) with the expressions $x_k^{\mathrm{T}}Qx_k + u_k^{\mathrm{T}}Ru_k$ for $Q, R > 0$ in [13, 15, 18–22, 24, 26, 32] satisfy this property. Moreover, all the utility functions with the expressions $R(x_k) + W(u_k)$ for $Q(x_k), W(u_k) > 0$ in [14, 16] also satisfy this property. Therefore the criterion in (62) is not strong and can easily be satisfied for real optimal control systems. In this paper, to release the computation quantity, we adopt the utility that satisfies (61).

From Theorems 5 and 6 we can see that the information of the parameter γ_{i+1} should be used while the value of γ_{i+1} is usually difficult to obtain. So in the following part, we give a more simplified convergence justification theorem of the iterative ADP algorithm.

Theorem 7: Let the iterative performance index function $\hat{\mathcal{V}}_i(x_k, u(p_1^i, \ldots, p_m^i))$ and the numerical iterative control $u(p_1^i, \ldots, p_m^i)$ be obtained by (28) and (29), respectively. Let $\bar{\epsilon}_{i+1}$ be expressed as in (46). For $\forall i = 0, 1, \ldots$, if the utility function $U(x_k, u_k)$ satisfies (61) and the iterative approximation error satisfies

$$\begin{aligned}\bar{\epsilon}_{i+1}(x_k) \leq \hat{\mathcal{V}}_i(x_k, u(p_1^i, \ldots, p_m^i)) - \frac{\hat{\mathcal{V}}_0(x_k, 0)}{\hat{\mathcal{V}}_0(x_k, 0) - U(x_k, 0)} \\ \times (\hat{\mathcal{V}}_i(x_k, u(p_1^i, \ldots, p_m^i)) - U(x_k, 0))\end{aligned} \tag{64}$$

then we have the numerical iterative control law $u(p_1^i, \ldots, p_m^i)$ stabilises the non-linear system (1) and simultaneously makes the iterative performance index function $\hat{\mathcal{V}}_{i+1}(x_k, u(p_1^i, \ldots, p_m^i))$ converge to a finite neighbourhood of $J^*(x_k)$, as $i \to \infty$.

Proof: First, we look at (19) and (20). Without loss of generality, we let $\tilde{\gamma}(x_k) = (J^*(x_k) - U(x_k, u^*(x_k))/U(x_k, u^*(x_k)))$ and $\tilde{\delta}(x_k) = (V_0(x_k)/J^*(x_k))$. Then, we can obtain $\tilde{\gamma}(x_k)$ $\tilde{\delta}(x_k) = (V_0(x_k)/U(x_k, u^*(x_k))) - \tilde{\delta}(x_k)$. According to (56), we can obtain $\max_{x_k\in\mathbb{R}^n, i=0,1,\ldots}\{\delta_{i+1}(x_k)\} \leq \max_{x_k\in\mathbb{R}^n}\tilde{\delta}(x_k)$, and $\max_{x_k\in\mathbb{R}^n, i=0,1,\ldots}\{(\hat{\mathcal{V}}_0(x_k, 0)/U(x_k, 0)) - \tilde{\delta}_{i+1}(x_k)\} \geq \max_{x_k\in\mathbb{R}^n}\{\tilde{\gamma}(x_k)\tilde{\delta}(x_k)\} = \gamma\delta$. Next, we note that if $\bar{\sigma}_{i+1}(x_k)$ satisfies

$$\bar{\sigma}_{i+1}(x_k) \leq 1 + \frac{(\tilde{\delta}_{i+1}(x_k) - 1)U(x_k, 0)}{\hat{\mathcal{V}}_0(x_k, 0) - \tilde{\delta}_{i+1}(x_k)U(x_k, 0)} \tag{65}$$

then we have

$$\begin{aligned}\sigma &= \max_{x_k\in\mathbb{R}^n, i=0,1,\ldots} \bar{\sigma}_{i+1}(x_k) \\ &\leq \max_{x_k\in\mathbb{R}^n, i=0,1,\ldots}\left\{1 + \frac{(\tilde{\delta}_{i+1}(x_k) - 1)U(x_k, 0)}{\hat{\mathcal{V}}_0(x_k, 0) - \tilde{\delta}_{i+1}(x_k)}U(x_k, 0)\right\} \\ &\leq 1 + \frac{\delta - 1}{\gamma\delta}\end{aligned} \tag{66}$$

holds. Putting (55) and (56) into (65), we can obtain (64). The proof is completed. □

Remark 6: From Theorem 7, we can see that the parameter γ_{i+1} is omitted in (64) and this makes the convergence criterion much simpler. This is the significant merit of this method, which is an important property we must point out. From (65) we can see that for $\forall\, i = 0, 1, \ldots$, the right-hand side of the inequality is not necessarily larger than 1. This means the approximation error $\bar{\epsilon}_{i+1}$ may be smaller than zero, which makes the convergence criterion (64) invalid, which is a shortcoming of this justification method. In Section 5, we will give a simulation example to show this property. To summarise, we recommend to verify the convergence property by (62) to guarantee the effectiveness of the convergence criterion.

3.3 Summary of the numerical iterative θ-ADP algorithm

Now, we summarise the numerical iterative θ-ADP algorithm.

Step 1. Choose randomly an array of initial states x_0 and choose the approximation precision ζ. Give the set of NCs $\mathfrak{A}$ by the expression (27). Give the max iteration index $i_{\max}$.

Step 2. Choose a large enough θ. Let $i = 0$ and $\hat{V}_0(x_k) = \theta\Psi(x_k)$, where $\Psi(x_k) \in \bar{\Psi}_{x_k}$.

Step 3. According to $\mathfrak{A}$, implement the numerical iterative θ-ADP algorithm (8)–(9) to obtain $\hat{v}_0(x_k)$ and $\hat{V}_1(x_k)$.

Step 4. Obtain $u(p_1^0, \ldots, p_m^0) = \hat{v}_0(x_k)$ by (28) and obtain $\bar{\mathfrak{A}}(p_1^0, p_2^0, \ldots, p_m^0)$ by (30). Obtain $\hat{\mathcal{V}}_1(x_k, u(p_1^0, \ldots, p_m^0))$ by (29).

Step 5. Solve the Lipschitz constant $L_0(p_1^0, p_2^0, \ldots, p_m^0)$ according to (50). Compute $\bar{\epsilon}_1(x_k)$ by (46).

Step 6. If $\bar{\epsilon}_1(x_k)$ satisfies (62), then let $i = i + 1$ and go to next step; otherwise, go to Step 11.

Step 7. For $i = 1, 2, \ldots$, implement the numerical iterative θ-ADP algorithm (10)–(11) to obtain $\hat{v}_i(x_k)$ and $\hat{V}_{i+1}(x_k)$. Obtain $u(p_1^i, \ldots, p_m^i) = \hat{v}_i(x_k)$ by (28) and obtain $\bar{\mathfrak{A}}(p_1^i, \ldots, p_m^i)$ by (30). Obtain $\hat{\mathcal{V}}_{i+1}(x_k, u(p_1^i, \ldots, p_m^i))$ by (29).

Step 8. Obtain the Lipschitz constant $L_i(p_1^i, \ldots, p_m^i)$ according to (50). Compute $\bar{\epsilon}_{i+1}(x_k)$ by (46).

Step 9. If $\bar{\epsilon}_{i+1}(x_k)$ satisfies (62), then go to next step. Otherwise, go to Step 11.

Step 10. If $|\hat{V}_{i+1}(x_k) - \hat{V}_i(x_k)| \leq \zeta$, then the iterative performance index function is converged and go to Step 11; else if $i > i_{\max}$, then go to Step 11; else, let $i = i + 1$ and go to Step 7.

Step 11. Stop.

4 Simulation studies

To evaluate the performance of our numerical iterative θ-ADP algorithm, we choose two examples with quadratic utility function for numerical experiments.

Example 1: Our first example is chosen as the example in [22, 32, 35, 36]. We consider the following system

$$x_1(k+1) = [x_1^2(k) + x_2^2(k) + u(k)]\cos(x_2(k))$$
$$x_2(k+1) = [x_1^2(k) + x_2^2(k) + u(k)]\sin(x_2(k)) \qquad (67)$$

Let $x_k = [x_1(k), x_2(k)]^{\mathrm{T}}$ denote the system state vector and $u_k = u(k)$ denote the control. Let $\mathfrak{A} = \{-2, -2+\varsigma, -2+2\varsigma, \ldots, 2\}$, where ς is the grid step. The performance index function is defined as (4) with the utility function $U(x_k, u_k) = x_k^{\mathrm{T}} Q x_k + u_k^{\mathrm{T}} R u_k$, where $Q = R = I$ and I is the identity matrix with suitable dimensions. The initial state is $x_0 = [1, -1]^{\mathrm{T}}$.

The iterative ADP algorithm runs for 30 iteration steps to guarantee the convergence of the iterative performance index function. The curves of the admissible approximation error obtained by (62) and (64) are displayed in Figs. 1 and 2, respectively. From Fig. 2, we can see that the for some states x_k, the admissible approximation error is smaller than zero,

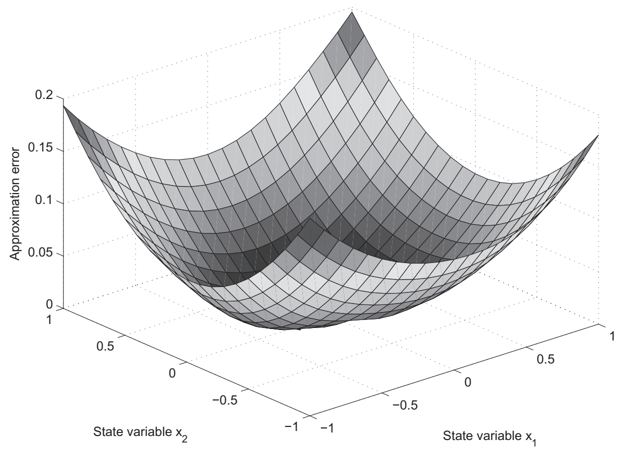

Fig. 1 *Curve of the admissible approximation error obtained by (62)*

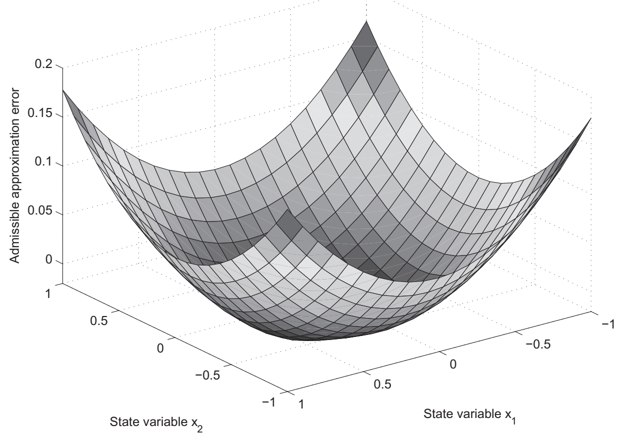

Fig. 2 *Curve of the admissible approximation error obtained by (64)*

which makes the convergence criterion (64) invalid. While from Fig. 1, we can see that the admissible approximation error curve is above zero which makes the convergence criterion (62) effective for all x_k.

To show the effectiveness of the numerical iterative ADP algorithm, we choose four different grid steps. Let $\varsigma = 10^{-8}$, 10^{-4}, 10^{-2}, 10^{-1}, respectively. The trajectories of the iterative performance index function are shown in Figs. 3*a–d*, respectively. For $\varsigma = 10^{-8}$ and $\varsigma = 10^{-4}$, implement the approximate optimal control for system (67), respectively. Let the implementation time be $T_f = 40$. The trajectories of the states and controls are displayed in Figs. 4*a–d*, respectively. When $\varsigma = 10^{-2}$, we can see that the iterative performance index function is not completely converged within 30 iteration steps. The trajectories of the state are displayed in Fig. 5*a* and the corresponding control trajectory is displayed in Fig. 5*b*. When $\varsigma = 10^{-1}$, we can see that the iterative performance index functions is not convergent. The control system is not stable.

In this paper, it is shown that if the inequality (57) holds, then for $\forall\, i = 0, 1, \ldots$, the numerical iterative control $\hat{v}_i(x_k)$ stabilises the system (67), which makes the numerical iterative θ-ADP algorithm implementable both on-line and off-line. In Figs. 6*a–d*, we give the system state and control trajectories of the system (67) under the iterative control law $\hat{v}_0(x_k)$ with $\varsigma = 10^{-8}$ and $\varsigma = 10^{-4}$, respectively. In Figs. 7*a–d*, we give the system state and control trajectories of the system (67) under the iterative control law $\hat{v}_0(x_k)$ with $\varsigma = 10^{-2}$ and $\varsigma = 10^{-1}$, respectively.

Remark 7: In [32, 33], only the convergence of the iterative performance index function is considered, while the convergence of the iterative performance index function cannot guarantee the stability of the system. From Fig. 3*c*, for $\varsigma = 10^{-2}$, we can see that the iterative performance index function is convergent. From Figs. 7*a* and *b* we can see that the system is not stable under the iterative control $\hat{v}_0(x_k)$. This makes the iterative ADP algorithm in [32, 33] is only implementable off-line. While from the simulation results such as Figs. 6*a–d*, we can see that if the approximation error satisfies (62) the stability of the system under the numerical controller can be satisfied, which makes the numerical iterative θ-ADP algorithm implementable both on-line and off-line. This is an obvious advantage of the present algorithm in this paper.

Example 2: We now examine the performance of the present algorithm in a practical torsional pendulum system [37]. The dynamics of the pendulum is given as follows

$$\begin{cases} \dfrac{\mathrm{d}\theta}{\mathrm{d}t} = \omega \\ J\dfrac{\mathrm{d}\omega}{\mathrm{d}t} = u - Mgl\sin\theta - f_d\dfrac{\mathrm{d}\theta}{\mathrm{d}t} \end{cases} \qquad (68)$$

where $M = 1/3\,\mathrm{kg}$ and $l = 2/3\,\mathrm{m}$ are the mass and length of the pendulum bar, respectively. Let $J = 4/3Ml^2$ and $f_{\mathrm{d}} = 0.2$ be the rotary inertia and frictional factor, respectively. Let $g = 9.8$ be the gravity acceleration. Discretising the system function and the performance index function using Euler method [38] with the sampling interval $\Delta t = 0.1\mathrm{s}$ leads to

$$\begin{bmatrix} x_{1(k+1)} \\ x_{2(k+1)} \end{bmatrix} = \begin{bmatrix} 0.1x_{2k} + x_{1k} \\ -0.49\sin(x_{1k}) - 0.1f_{\mathrm{d}} \times x_{2k} + x_{2k} \end{bmatrix} + \begin{bmatrix} 0 \\ 0.1 \end{bmatrix} u_k \qquad (69)$$

IET Control Theory Appl., 2013, Vol. 7, Iss. 11, pp. 1472–1486
doi: 10.1049/iet-cta.2012.0486

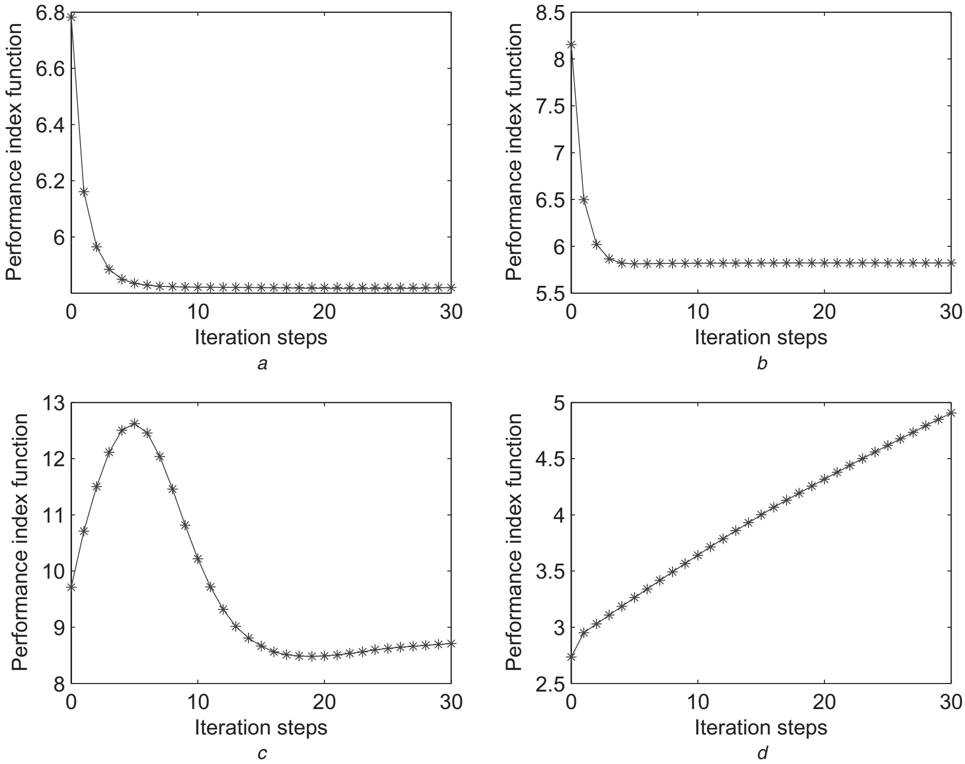

Fig. 3 *Trajectories of the iterative performance index functions*
a $\varsigma = 10^{-8}$
b $\varsigma = 10^{-4}$
c $\varsigma = 10^{-2}$
d $\varsigma = 10^{-1}$

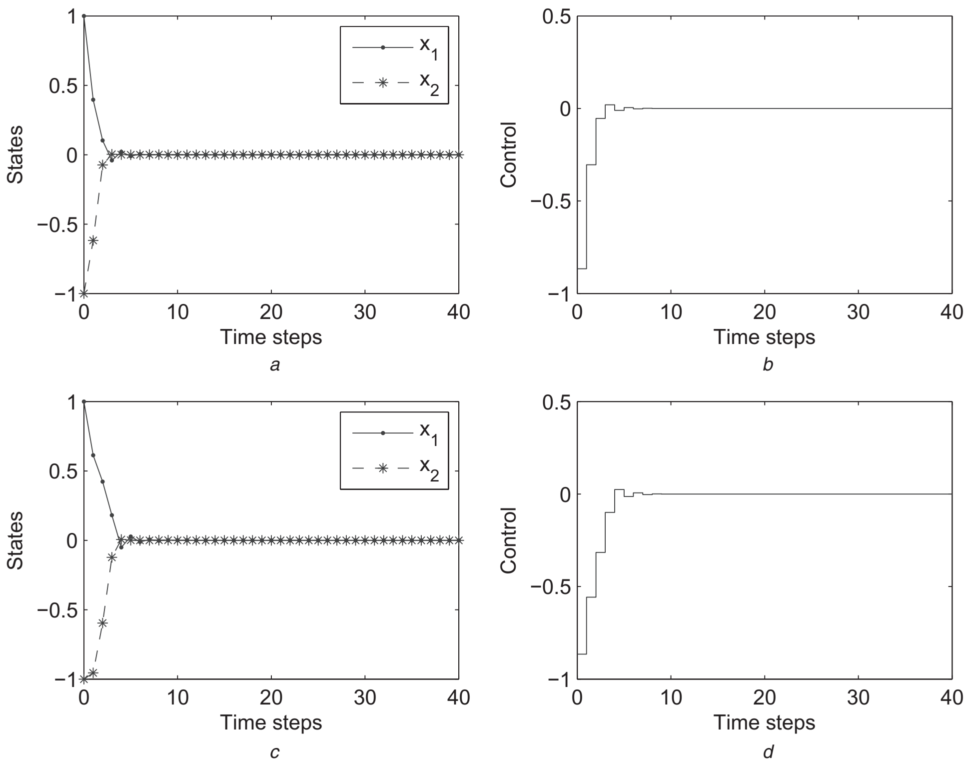

Fig. 4 *Control and state trajectories*
a State trajectories for $\varsigma = 10^{-8}$
b Control trajectory for $\varsigma = 10^{-8}$
c State trajectories for $\varsigma = 10^{-4}$
d Control trajectory for $\varsigma = 10^{-4}$

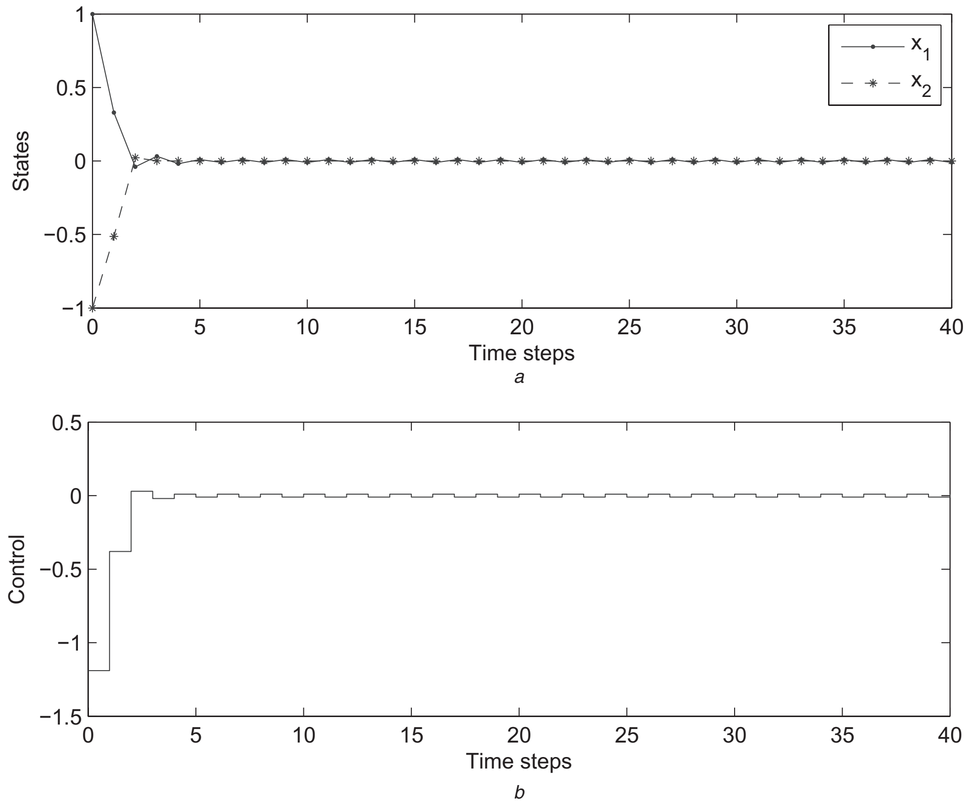

Fig. 5 *Control and state trajectories*
a State trajectories for $\varsigma = 10^{-2}$
b Control trajectory for $\varsigma = 10^{-2}$

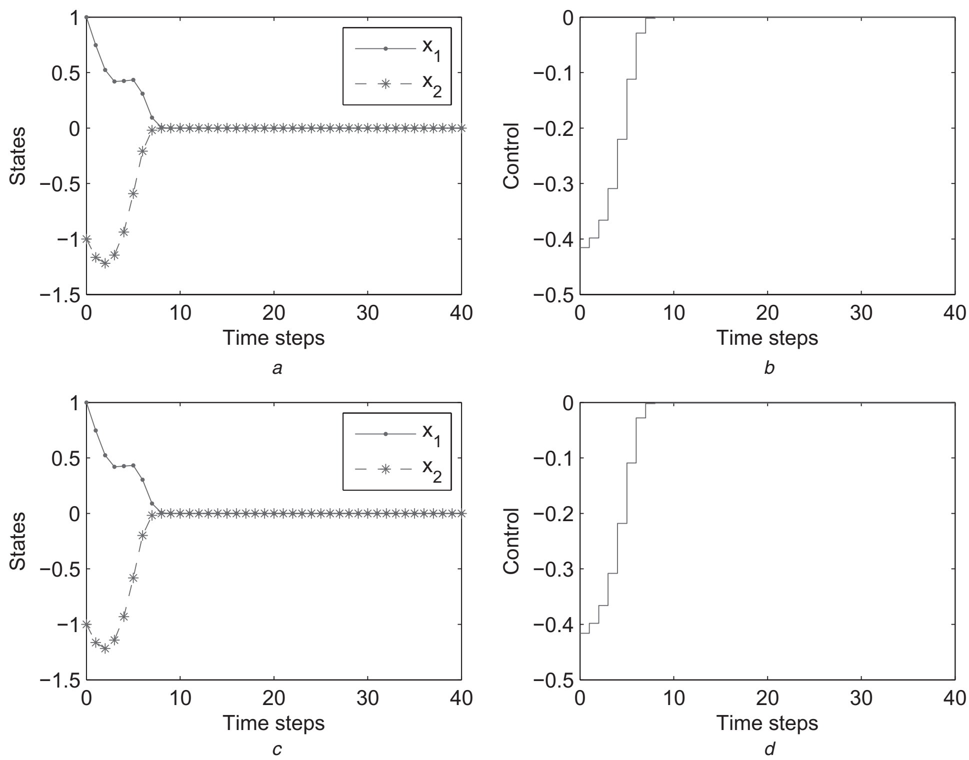

Fig. 6 *Control and state trajectories under* $\hat{v}_0(x_k)$
a State trajectories for $\varsigma = 10^{-8}$
b Control trajectory for $\varsigma = 10^{-8}$
c State trajectories for $\varsigma = 10^{-4}$
d Control trajectory for $\varsigma = 10^{-4}$

IET Control Theory Appl., 2013, Vol. 7, Iss. 11, pp. 1472–1486
doi: 10.1049/iet-cta.2012.0486

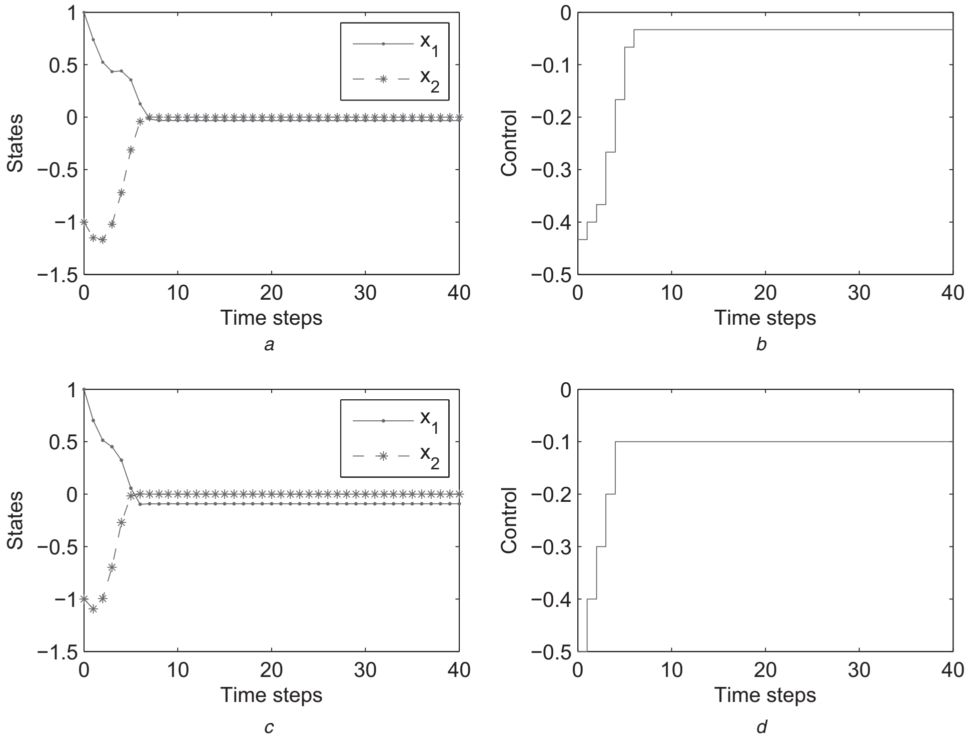

Fig. 7 *Control and state trajectories under $\hat{v}_0(x_k)$*

a State trajectories for $\varsigma = 10^{-2}$
b Control trajectory for $\varsigma = 10^{-2}$
c State trajectories for $\varsigma = 10^{-1}$
d Control trajectory for $\varsigma = 10^{-1}$

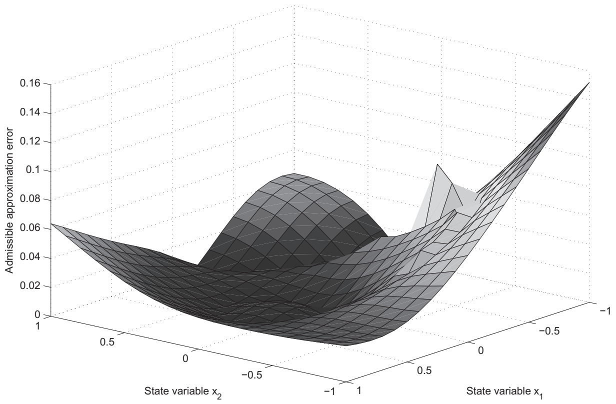

Fig. 8 *Curve of the admissible approximation error obtained by (62)*

where $x_{1k} = \theta_k$ and $x_{2k} = \omega_k$. Assume that the sampling interval satisfies the Shannon's sampling theorem [39]. Let the initial state be $x_0 = [1, -1]^{\mathrm{T}}$. The utility function is chosen the same as the one in Example 1.

The numerical iterative θ-ADP algorithm runs for 16 iteration steps to guarantee the convergence of the iterative performance index function. The curve of the admissible approximation error obtained by (62) is displayed in Fig. 8.

Let the grid step of the control be $\varsigma = 10^{-6}$ to guarantee the convergence condition (62). The convergence trajectory of the iterative performance index function is shown in Fig. 9*a*. To illustrate the performance of the present numerical iterative θ-ADP algorithm, our results will be compared with traditional value iteration algorithm, which is widely used in [20, 21, 27, 28]. Let the initial performance index function be $V_0(x_k) \equiv 0$. We implement the value iteration algorithm for 30 iterations. The convergence trajectory obtained value iteration algorithm is shown in Fig. 9*b*.

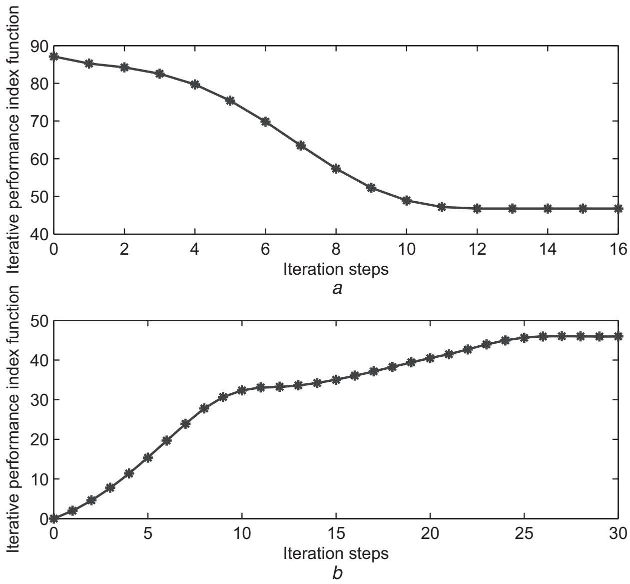

Fig. 9 *Convergence trajectories of the iterative performance index functions*

a Trajectory obtained by numerical iterative θ-ADP algorithm
b Trajectory obtained by value iteration algorithm

Remark 8: From Figs. 9*a* and *b* we can see that the two iterative performance index functions obtained by the numerical

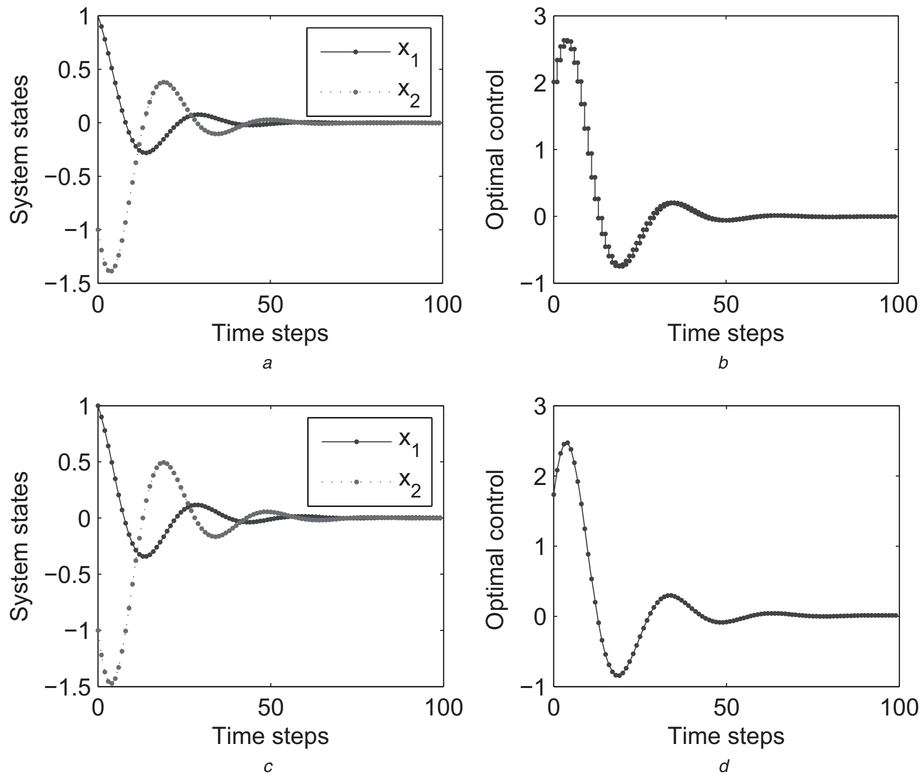

Fig. 10 *Optimal control and state trajectories*

a Optimal state trajectories obtained by numerical iterative θ-ADP algorithm
b Optimal control trajectory obtained by numerical iterative θ-ADP algorithm
c Optimal state trajectories obtained by value iteration algorithm
d Optimal control trajectory obtained by value iteration algorithm

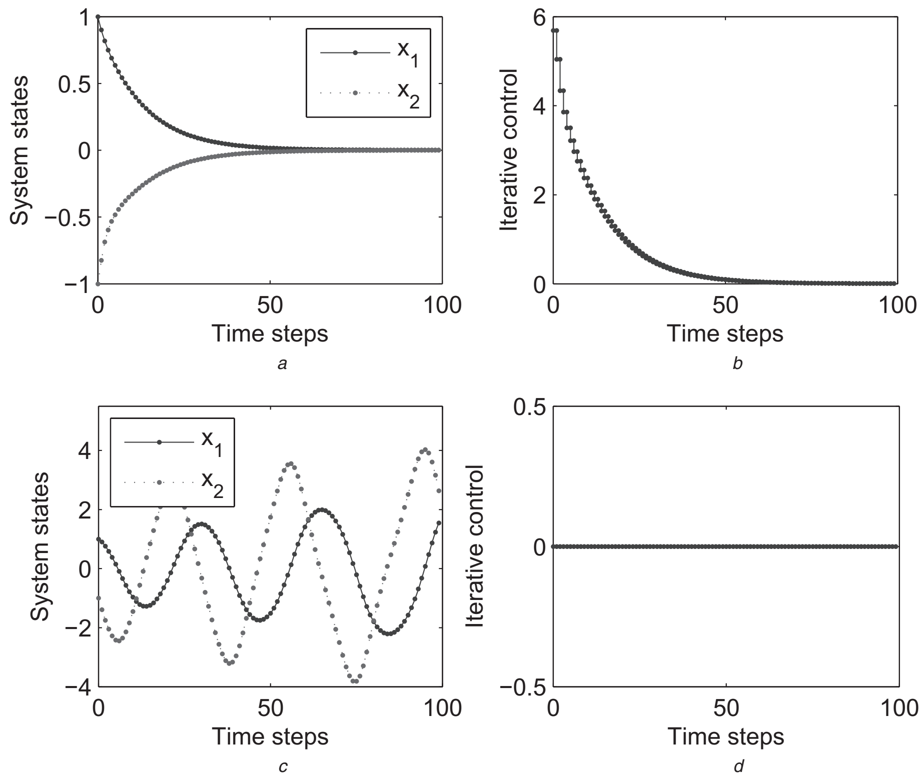

Fig. 11 *Iterative state and control trajectories under the first iteration control law*

a State trajectories under $\hat{v}_0(x_k)$ obtained by numerical iterative θ-ADP algorithm
b First iteration control trajectory $\hat{v}_0(x_k)$ by numerical iterative θ-ADP algorithm
c State trajectories under $\bar{v}_0(x_k)$ obtained by value iteration algorithm
d First iteration control trajectory $\bar{v}_0(x_k)$ by value iteration algorithm

doi: 10.1049/iet-cta.2012.0486

iterative θ-ADP algorithm and the value iteration algorithm are both convergent, while the convergence properties of the two iterative performance index functions are different. For the numerical iterative θ-ADP algorithm, the iterative performance index function is monotonically non-increasing and convergent. However, for the value iteration algorithm, the iterative performance index function is monotonically non-decreasing and convergent. This is an obvious difference between the two algorithms because of the different choice of the initial performance index function.

The optimal state and control trajectories by the numerical iterative θ-ADP algorithm are shown in Figs. 10*a* and *b*, respectively. The optimal state and control trajectories by the traditional value iteration algorithm are shown in Figs. 10*c* and *d*, respectively.

From the simulation results, we can see that the numerical iterative θ-ADP algorithm effectively obtains the optimal control law of the torsional pendulum system. On the other hand, we point out that for value iteration, it cannot make the system stable under the iterative control law, while for the numerical iterative θ-ADP algorithm, the stability of the system can be guaranteed. In Fig. 11*a*, we give the system state trajectories of the system (69) under the iterative control law $\hat{v}_0(x_k)$ by numerical iterative θ-ADP algorithm. The corresponding control trajectory is shown in Fig. 11*b*. Let $\bar{v}_0(x_k)$ be the control law of the first iteration obtained by value iteration algorithm. Then the system state trajectories of the system (69) under the iterative control law $\bar{v}_0(x_k)$ by value iteration algorithm are shown in Fig. 11*c*. The corresponding control trajectory is shown in Fig. 11*d*.

Remark 9: From the simulation results, we see that the stability of the system under the iterative control law obtained by the value iteration algorithm cannot be guaranteed. This makes the value iteration algorithms can only implemented off-line to obtain the optimal control law. For our present numerical iterative θ-ADP algorithm, it is successfully obtain the optimal control law for the torsional pendulum system. If the approximation error is satisfied, then the iterative performance index function is convergent to the finite neighbourhood of the optimal performance index functions. We emphasise that the stability of the system is also guaranteed. This is an great advantage of the present algorithm in this paper comparing the iteration algorithms.

5 Conclusions

In this paper, we have developed an effective numerical iterative θ-ADP algorithm to find the infinite horizon optimal control for discrete-time non-linear systems. In the numerical iterative θ-ADP algorithm, any of the iterative control is stable for the non-linear system which means the present algorithm can be used both on-line and off-line. Convergence analysis of the performance index function for the iterative ADP algorithm is proved and the stability proofs are also given. Finally, two simulation examples are given to illustrate the performance of the present algorithm.

6 Acknowledgments

This work was supported in part by the National Natural Science Foundation of China under Grants 61034002, 61233001, 61273140, in part by Beijing Natural Science Foundation under Grant 4132078, and in part by the Early Career Development Award of SKLMCCS.

7 References

1 Padhi, R., Balakrishnan, S.N.: 'Optimal dynamic inversion control design for a class of nonlinear distributed parameter systems with continuous and discrete actuators', *IET Control Theory Appl.*, 2007, **1**, (6), pp. 1662–1671

2 Meng, D., Jia, Y., Du, J.: 'Robust iterative learning control design for uncertain time-delay systems based on a performance index', *IET Control Theory Appl.*, 2010, **4**, (5), pp. 759–772

3 Jiang, Q., Xi, H.S., Yin, B.Q.: 'Adaptive optimisation of timeout policy for dynamic power management based on semi-Markov control processes', *IET Control Theory Appl.*, 2010, **4**, (10), pp. 1945–1958

4 Iyasere, E., Salah, M., Dawson, D., Wagner, J., Tatlicioglu E.: 'Optimum seeking-based non-linear controller to maximise energy capture in a variable speed wind turbine', *IET Control Theory Appl.*, 2012, **6**, (4), pp. 526–532

5 Lewis, F.L., Syrmos, V.L.: 'Optimal control' (John Wiley, 1995)

6 Kushner, H.J.: 'Numerical algorithms for optimal controls for non-linear stochastic systems with delays', *IEEE Trans. Autom. Control*, 2010, **55**, (9), pp. 2170–2176

7 Younkin, G., Hesla, E.: 'Origin of numerical control', *IEEE Ind. Appl. Mag.*, 2008, **14**, (5), pp. 10–12

8 Zhang, C., Ordonez, R.: 'Numerical optimization-based extremum seeking control with application to ABS design', *IEEE Trans. Autom. Control*, 2007, **52**, (3), pp. 454–467

9 Werbos, P.J.: 'Advanced forecasting methods for global crisis warning and models of intelligence', *Gen. Syst. Yearb.*, 1977, **22**, pp. 25–38

10 Werbos, P.J.: 'A menu of designs for reinforcement learning over time', in Miller W.T., Sutton R.S., Werbos P.J. (Eds): 'Neural networks for control' (MIT Press, 1991), pp. 67–95

11 Li, X.D., Xiao, T.F., Zheng H.X.: 'Adaptive discrete-time iterative learning control for non-linear multiple input multiple output systems with iteration-varying initial error and reference trajectory', *IET Control Theory Appl.*, 2011, **5**, (9), pp. 1131–1139

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

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

14 Murray, J.J., Cox, C.J., Lendaris, G.G., Saeks, R.: 'Adaptive dynamic programming', *IEEE Trans. Syst. Man Cybern. C, Appl. Rev.*, 2002, **32**, (2), pp. 140–153

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

16 Abu-Khalaf, M., Lewis, F.L.: 'Nearly optimal control laws for non-linear systems with saturating actuators using a neural network HJB approach', *Automatica*, 2005, **41**, (5), pp. 779–791

17 Lin, W.S., Chang, L.H., Yang, P.C.: 'Adaptive critic anti-slip control of wheeled autonomous robot', *IET Control Theory Appl.*, 2007, **1**, (1), pp. 51–57

18 Lewis, F.L., Vrabie, D.: 'Reinforcement learning and adaptive dynamic programming for feedback control', *IEEE Circuits Syst. Mag.*, 2009, **9**, (3), pp. 32–50

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

20 Beard, R.: 'Improving the closed-loop performance of nonlinear systems'. PhD thesis, Rensselaer Polytechnic Institute, Troy, New York, USA, 1995

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

22 Wei, Q., Liu, D.: 'Adaptive dynamic programming with stable value iteration algorithm for discrete-time nonlinear systems'. Proc. Int. Joint Conf. Neural Networks, Brisbane, Australia, June 2012, pp. 1–6

23 Dierks, T., Jagannathan, S.: '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.*, 2012, **23**, (7), pp. 1118–1129

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

25 Wei, Q., Liu, D.: 'An iterative ϵ-optimal control scheme for a class of discrete-time nonlinear systems with unfixed initial state', *Neural Netw.*, 2012, **32**, pp. 236–244
26 Zhang, H., Wei, Q., Luo, Y.: '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.*, 2008, **38**, (4), pp. 937–942
27 Lincoln, B., Rantzer, A.: 'Relaxing dynamic programming', *IEEE Trans. Autom. Control*, 2006, **51**, (8), pp. 1249–1260
28 Zhang, H., Luo, Y., Liu, D.: 'The RBF neural network-based near-optimal control for a class of discrete-time affine nonlinear systems with control constraint', *IEEE Trans. Neural Netw.*, 2009, **20**, (9), pp. 1490–1503
29 Jin, N., Liu, D., Huang, T., Pang, Z.: 'Discrete-time adaptive dynamic programming using wavelet basis function neural networks'. Proc. IEEE Symp. Approximate Dynamic Programming and Reinforcement Learning, Honolulu, USA, April 2007, pp. 135–142
30 Liu, D., Wang, D., Zhao, D., Wei, Q., Jin, N.: 'Neural-network-based optimal control for a class of unknown discrete-time nonlinear systems using globalized dual heuristic programming', *IEEE Trans. Autom. Sci. Eng.*, 2012, **9**, (3), pp. 628–634
31 Yang, Q., Jagannathan, S.: 'Reinforcement learning controller design for affine nonlinear discrete-time systems using online approximators', *IEEE Trans. Syst. Man Cybern. B, Cybern.*, 2012, **42**, (2), pp. 377–390
32 Liu, D., Wei, Q.: 'Finite-approximation-error based optimal control approach for discrete-time nonlinear systems', *IEEE Trans. Syst. Man Cybern. B, Cybern.*, 2013, **43**, (2), pp. 779–789
33 Wei, Q., Liu, D.: 'A new optimal control method for discrete-time nonlinear systems with approximation errors'. Proc. World Congress on Intelligent Control and Automation, Beijing, China, July 2012, pp. 185–190
34 Agricola, I., Friendrich, T.: 'Elementary geometry' (American Mathematical Society, 2008)
35 Navarro-Lopez, E.M.: 'Local feedback passivation of nonlinear discrete-time systems through the speed-gradient algorithm', *Automatica*, 2007, **43**, (7), pp. 1302–1306
36 Sira-Ramirez, H.: 'Non-linear discrete variable structure systems in quasi-sliding mode', *Int. J. Control*, 1991, **54**, (5), pp. 1171–1187
37 Si, J., Wang, Y.T.: 'On-line learning control by association and reinforcement', *IEEE Trans. Neural Netw.*, 2001, **12**, (2), pp. 264–276
38 Padhi, R., Unnikrishnan, N., Wang, X., Balakrishnan, S.N.: 'A single network adaptive critic (SNAC) architecture for optimal control synthesis for a class of nonlinear systems', *Neural Netw.*, 2006, **19**, (10), pp. 1648–1660
39 Marks, I.I., Robert, J.: 'Introduction to Shannon sampling and interpolation theory' (Springer–Verlag, 1991)

IET Control Theory Appl., 2013, Vol. 7, Iss. 11, pp. 1472–1486
doi: 10.1049/iet-cta.2012.0486