

Chapter 2

An Iterative ϵ -Optimal Control Scheme for a Class of Discrete-Time Nonlinear Systems with Unfixed Initial State

2.1 Introduction

Strictly speaking, most real-world control systems need to be effectively controlled within finite time horizon (finite horizon for brief), such as stabilized within finite horizon. In many theoretical discussions, however, controllers are generally designed to make the controlled systems stabilized within infinite time horizon [3, 8, 21, 33]. The design of finite-time horizon controller faces a major obstacle in comparison with the infinite horizon one. For infinite horizon control problems, Lyapunov theory is popularly used and asymptotic results for the control systems are generally obtained [7, 32]. That is, the system cannot really be stabilized until the time reaches infinity. While for finite horizon control problems, the system must be stabilized to zero within finite time [9, 16, 19]. Due to the lack of methodology and the fact that the control step is difficult to determine, the controller design of finite horizon problems is still a challenge to control engineers. On the other hand, optimization is always an important objective for the design of control systems. This is the reason why optimal control has been paid much attention by many researchers for over 50 years and applied to many application domains [4, 5, 14, 17, 30].

Adaptive dynamic programming (ADP) algorithm was proposed by Werbos [28, 29], as a powerful methodology to solve optimal control problems forward-in-time. Though ADP algorithms have made great progress in the optimal control field [1, 6, 13, 15, 20, 24, 25, 27, 31], the discussions about finite horizon optimal control problems are scarce. To the best of our knowledge, only Wang et al. [23] discussed finite horizon optimal control problem with fixed initial state. Wei and Liu [24, 25] proposed an iterative ADP algorithm with unfixed initial state while it requires that the system can be reach zero in one step control to initialize the algorithm which limits its application very much. So, it is still an open problem how to solve the optimal control problem in finite horizon with unfixed initial state when the system cannot reach zero directly. This motivates our research.

In this chapter, for the first time, we will show how to find an approximate optimal control that makes the iterative value function converge to the greatest lower bound

of all performance indices within an error according to ϵ (called ϵ -error bound for brief) without the rigorous initial condition in [10–12, 24, 25]. It is also shown that the corresponding approximate optimal control (called ϵ -optimal control) can make the iterative value function converge to the ϵ -error bound within finite steps where the iterative ADP algorithm is initialized by an arbitrary admissible control sequence. In a brief, the main contributions of this chapter include:

- (1) Present a new proof that the iterative ADP algorithm can converge to the optimum initialized by an arbitrary admissible control sequence.
- (2) Prove that the ϵ -optimal control can make the iterative value function converge to the greatest lower bound of all performance indices within an error ϵ for unfixed initial state and the rigorous initial condition in [24, 25] is omitted.
- (3) Obtain the length of the ϵ -optimal control.

2.2 Problem Statement

In this chapter, we consider the following discrete-time nonlinear systems:

$$x_{k+1} = F(x_k, u_k), \quad k = 0, 1, 2, \dots, \quad (2.1)$$

where, $x_k \in \mathbb{R}^n$ is the state and $u_k \in \mathbb{R}^m$ is the control vector. Let $x_0 \in \Omega_0$ be the initial state where $\Omega_0 \subset \mathbb{R}^n$ is the domain of initial state. Let the system function $F(x_k, u_k)$ be continuous $\forall x_k, u_k$ and $F(0, 0) = 0$. We will study the optimal control problems for system (2.1) with finite horizon and unspecified terminal time. The performance index function for state x_0 under the control sequence $\underline{u}_0^{N-1} = (u_0, u_1, \dots, u_{N-1})$ is defined as

$$J(x_0, \underline{u}_0^{N-1}) = \sum_{k=0}^{N-1} U(x_k, u_k), \quad (2.2)$$

where $U(x_k, u_k) \geq 0, \forall x_k, u_k$, is the utility function.

Let $\underline{u}_0^{N-1} = (u_0, u_1, \dots, u_{N-1})$ be a finite sequence of controls. We call the number of elements in the control sequence $\underline{u}_0^{N-1}$ the length of $\underline{u}_0^{N-1}$. Then $|\underline{u}_0^{N-1}| = N$. We denote the final state of the trajectory as $x^{(f)}(x_0, \underline{u}_0^{N-1})$, i.e., $x^{(f)}(x_0, \underline{u}_0^{N-1}) = x_N$. For all $k \geq 0$, the finite control sequence can be written as $\underline{u}_k^{k+i-1} = (u_k, u_{k+1}, \dots, u_{k+i-1})$ where $i \geq 1$. The final state can be written as $x^{(f)}(x_k, \underline{u}_k^{k+i-1})$ where $x^{(f)}(x_k, \underline{u}_k^{k+i-1}) = x_{k+i}$. Let $\underline{u}_k$ be an arbitrary finite horizon admissible control sequence starting at k . Let $\mathfrak{A}_{x_k} = \{\underline{u}_k : x^{(f)}(x_k, \underline{u}_k) = 0\}$ and

$$\mathfrak{A}_{x_k}^{(i)} = \{\underline{u}_k^{k+i-1} : x^{(f)}(x_k, \underline{u}_k^{k+i-1}) = 0, \quad |\underline{u}_k^{k+i-1}| = i\}$$

be the set of all finite horizon admissible control sequences of x_k with length i . Then, $\mathfrak{A}_{x_k} = \cup_{1 \leq i < \infty} \mathfrak{A}_{x_k}^{(i)}$. By this notation, a state x_k is controllable if and only if $\mathfrak{A}_{x_k} \neq \emptyset$. Define the optimal performance index function as

$$J^*(x_k) = \min_{\underline{u}_k} \{J(x_k, \underline{u}_k) : \underline{u}_k \in \mathfrak{A}_{x_k}\}. \quad (2.3)$$

Then, according to Bellman's principle of optimality [2], $J^*(x_k)$ satisfies the discrete-time HJB equation

$$J^*(x_k) = \min_{u_k} \{U(x_k, u_k) + J^*(F(x_k, u_k))\}. \quad (2.4)$$

and the law of optimal control vector is given by

$$u^*(x_k) = \arg \min_{u_k} \{U(x_k, u_k) + J^*(F(x_k, u_k))\}. \quad (2.5)$$

2.3 Properties of the Iterative Adaptive Dynamic Programming Algorithm

In this section, a new iterative ADP algorithm is developed to obtain the finite horizon optimal controller for nonlinear systems. The goal of the present iterative ADP algorithm is to construct an optimal control law $u^*(x_k)$, $k = 0, 1, \dots$, which drives the system from an arbitrary initial state x_0 to the singularity 0 within finite time, and simultaneously minimizes the performance index function. Convergence proofs will also be given to show that the iterative value function converges to the optimum.

2.3.1 Derivation of the Iterative ADP Algorithm

In the iterative ADP algorithm, the iterative value function and control law are updated by recurrent iterations, with the iteration index number i increasing from 0. Since x_k is controllable, there exists a finite horizon admissible control sequence $\underline{u}_k^{k+i-1} = \{u_k, u_{k+1}, \dots, u_{k+i-1}\} \in \mathfrak{A}_{x_k}^{(i)}$ that makes $x^{(f)}(x_k, \underline{u}_k^{k+i-1}) = x_{k+i} = 0$. Let $\underline{\nu}_k^{N-1} = \{\nu_k, \nu_{k+1}, \dots, \nu_{N-1}\}$ be an arbitrary admissible sequence in $\mathfrak{A}_{x_k}^{(N-k)}$, where N is an unspecified terminal time. Define the value function $\Phi(x_k)$ is the value function constructed by $\underline{\nu}_k^{N-1}$ which can be expressed by

$$\Phi(x_k) = J(x_k, \underline{\nu}_k^{N-1}). \quad (2.6)$$

Let $V_0(x_k) = \Phi(x_k)$ and the iterative value function $V_1(x_k)$ can be updated as

$$\begin{aligned} V_1(x_k) &= \min_{u_k} \{U(x_k, u_k) + V_0(x_{k+1})\} \\ &= U(x_k, v_1(x_k)) + V_0(F(x_k, v_1(x_k))), \end{aligned} \quad (2.7)$$

where the iterative control law $v_1(x_k)$ is computed as

$$\begin{aligned} v_1(x_k) &= \arg \min_{u_k} \{U(x_k, u_k) + V_0(x_{k+1})\} \\ &= \arg \min_{u_k} \{U(x_k, u_k) + V_0(F(x_k, u_k))\}, \end{aligned} \quad (2.8)$$

For $i = 1, 2, \dots$, the iterative ADP algorithm will iterate between the iterative value function

$$\begin{aligned} V_i(x_k) &= \min_{u_k} \{U(x_k, u_k) + V_{i-1}(x_{k+1})\} \\ &= U(x_k, v_i(x_k)) + V_{i-1}(F(x_k, v_i(x_k))), \end{aligned} \quad (2.9)$$

where the iterative control law $v_i(x_k)$ is computed as

$$\begin{aligned} v_i(x_k) &= \arg \min_{u_k} \{U(x_k, u_k) + V_{i-1}(x_{k+1})\} \\ &= \arg \min_{u_k} \{U(x_k, u_k) + V_{i-1}(F(x_k, u_k))\}. \end{aligned} \quad (2.10)$$

Remark 2.1 There exists a great difference between the iterative ADP algorithm (2.7)–(2.10) and the iterative ADP algorithm proposed in [24, 25]. In [24, 25], it requires the condition that for all $x_k \in \mathbb{R}^n$, there exists a control u_k that makes $F(x_k, u_k) = 0$ to initialize the iterative ADP algorithm. As is known, for general control systems, especially for nonlinear systems, there does not exist a control that makes $F(x_k, u_k) = 0$ hold for all x_k . So the initial condition for the iterative ADP algorithm in [24, 25] limits its application very much. In this chapter, the rigorous constrain in [24, 25] is omitted and the present iterative ADP algorithm begins with a value function constructed by an arbitrary finite horizon admissible control sequence. Thus, we can say that the present iterative ADP algorithm in this chapter is more effective.

Theorem 2.1 *Let x_k be an arbitrary state. Let the iterative value function $V_i(x_k)$ be obtained according to (2.7)–(2.10), and then we have*

$$\begin{aligned} V_i(x_k) &= \sum_{j=0}^{i-1} U(x_{k+j}, v_{i-j}(x_{k+j})) + \Phi(x_{k+i}) \\ &= \min_{u_k^{k+i-1}} \left\{ \sum_{j=0}^{i-1} U(x_{k+j}, u_{k+j}) \right\} + \Phi(x_{k+i}). \end{aligned} \quad (2.11)$$

Proof For $i = 0, 1, \dots$, according to the definition of $V_i(x_k)$ in (2.7)–(2.9), we can get

$$\begin{aligned}
V_i(x_k) &= \min_{u_k} \{U(x_k, u_k) + V_{i-1}(x_{k+1})\} \\
&= \min_{u_k} \left\{ U(x_k, u_k) + \min_{u_{k+1}} \left\{ U(x_{k+1}, u_{k+1}) \right. \right. \\
&\quad \left. \left. + \cdots + \min_{u_{k+i-1}} \{U(x_{k+i-1}, u_{k+i-1}) \right. \right. \\
&\quad \left. \left. + V_0(x_{k+i}) \} \cdots \right\} \right\},
\end{aligned}$$

where $V_0(x_{k+i}) = \Phi(x_{k+i})$. According to the optimality principle, we obtain

$$\begin{aligned}
V_i(x_k) &= \min_{\underline{u}_k^{k+i-1}} \{U(x_k, u_k) + U(x_{k+1}, u_{k+1}) + U(x_{k+2}, u_{k+2}) \\
&\quad + \cdots + U(x_{k+i-1}, u_{k+i-1}) + V_0(x_{k+i})\} \\
&= \min_{\underline{u}_k^{k+i-1}} \left\{ \sum_{j=0}^{i-1} U(x_{k+j}, u_{k+j}) \right\} \\
&\quad + \Phi(x_{k+i}).
\end{aligned} \tag{2.12}$$

According to (2.9), we have

$$V_i(x_k) = \sum_{j=0}^{i-1} U(x_{k+j}, v_{i-j}(x_{k+j})) + \Phi(x_{k+i}). \tag{2.13}$$

The proof is complete.

We can see that the HJB equation (2.4) is changed to a sequence of the iterative value function $V_i(x_k)$. From Theorem 2.1, $V_i(x_k)$ can be obtained by solving a optimal control problem with terminal constraint. Obviously, $V_i(x_k)$ is not necessary satisfies the HJB equation (2.4). Convergence analysis of the iterative value function $V_i(x_k)$ is required.

2.3.2 Properties of the Iterative ADP Algorithm

In the above subsection, we can see that the performance index function $J^*(x_k)$ solved by HJB equation (2.4) is replaced by a sequence of the iterative value functions $V_i(x_k)$ and the optimal control law $u^*(x_k)$ is replaced by a sequence of control laws $v_i(x_k)$, where $i \geq 1$ is the index of iteration. We can prove that $J^*(x_k)$ defined in (2.3) is the limit of $V_i(x_k)$ as $i \rightarrow \infty$.

Lemma 2.1 *Let the iterative value function $V_i(x_k)$ be defined by (2.9). Let $\underline{\mu}_k = (\mu_k, \mu_{k+1}, \dots) \in \mathfrak{A}_{x_k}$ be an arbitrary admissible control sequence. Define a new value function $P_i(x_k)$ as*

$$P_i(x_k) = U(x_k, \mu_k) + P_{i-1}(x_{k+1}), \quad (2.14)$$

with $P_0(x_k) = V_0(x_k) = \Phi(x_k)$, then we have $V_i(x_k) \leq P_i(x_k)$, $\forall i = 0, 1, \dots$

Theorem 2.2 *Let x_k be an arbitrary state vector and $V_0(x_k) = \Phi(x_k)$ is defined by (2.6). Then, the iterative value function $V_i(x_k)$ obtained by (2.7)–(2.10) is a monotonically nonincreasing sequence for all $i \geq 1$, i.e., $V_{i+1}(x_k) \leq V_i(x_k)$ for all $i = 0, 1, \dots$*

Proof We prove this conclusion by mathematical induction. First, we let $i = 0$.

Let $\underline{\mu}_k \in \mathfrak{A}_{x_k}$ be an arbitrary admissible control sequence. Define the value function $P_i(x_k)$ as (2.14). For $i = 0$, we have

$$\begin{aligned} P_1(x_k) &= U(x_k, \mu_k) + P_0(x_{k+1}) \\ &= U(x_k, \mu_k) + \Phi(x_{k+1}). \end{aligned} \quad (2.15)$$

According to the definition of $\Phi(x_k)$ in (2.6), we have

$$\Phi(x_k) - \Phi(F(x_k, \nu_k)) = J(x_k) - J(x_{k+1}) = U(x_k, \nu_k). \quad (2.16)$$

As $\underline{\mu}_k \in \mathfrak{A}_{x_k}$ is arbitrary, let $\nu_k = \mu_k$, and then we can obtain

$$V_0(x_k) = \Phi(x_k) = U(x_k, \nu_k) + \Phi(F(x_k, \nu_k)) = P_1(x_k) \quad (2.17)$$

holds. On the other hand, according to Lemma 2.1, we have

$$\begin{aligned} V_1(x_k) &= \min_{u_k} \{U(x_k, u_k) + V_0(x_{k+1})\} \leq P_1(x_k) \\ &= U(x_k, \nu_k) + P_0(x_{k+1}) \leq V_0(x_k), \end{aligned} \quad (2.18)$$

which proves $V_0(x_k) \geq V_1(x_k)$.

Hence, the conclusion holds for $i = 0$. Assume that the conclusion holds for $i = l - 1$, where $l = 1, 2, \dots$. Then, for $i = l$, define the value function $P_l(x_k)$ as

$$\begin{aligned} P_l(x_k) &= U(x_k, \nu_{l-1}(x_k)) + U(x_{k+1}, \nu_{l-2}(x_{k+1})) \\ &\quad + \dots + U(x_{k+l-2}, \nu_1(x_{k+l-2})) + U(x_{k+l-1}, \mu(x_{k+l-1})) + P_0(x_{k+l}) \\ &= \sum_{j=0}^{l-2} U(x_{k+j}, \nu_{l-j-1}(x_{k+j})) \\ &\quad + U(x_{k+l-1}, \mu(x_{k+l-1})) + \Phi(x_{k+l}). \end{aligned} \quad (2.19)$$

For $i = 0, 1, \dots$, we have the iterative value function $V_l(x_k)$ can be expressed as

$$V_l(x_k) = \sum_{j=0}^{l-1} U(x_{k+j}, v_{l-j}(x_{k+j})) + \Phi(x_{k+l}). \quad (2.20)$$

According to (2.6), we have

$$\Phi(x_{k+l}) = U(x_{k+l}, \nu(x_{k+l})) + \Phi(F(x_{k+l}, \nu(x_{k+l}))). \quad (2.21)$$

Let $\nu_{k+l} = \mu_{k+l}$, and then according to (2.19) and (2.20), we can get

$$\begin{aligned} V_l(x_k) &= \sum_{j=0}^{l-1} U(x_{k+j}, v_{l-j-1}(x_{k+j})) + \Phi(x_{k+l}) \\ &= \sum_{j=0}^{l-1} U(x_{k+j}, v_{l-j-1}(x_{k+j})) + U(x_{k+l}, \nu_{k+l}) + \Phi(x_{k+l+1}) \\ &= P_{l+1}(x_k). \end{aligned} \quad (2.22)$$

According to Lemma 2.1, we have $V_{l+1}(x_k) \leq P_{l+1}(x_k)$. Therefore, we can obtain $V_{l+1}(x_k) \leq V_l(x_k)$.

Remark 2.2 For the iterative ADP algorithm proposed in [24, 25, 31], the iterative value function $V_i(x_k)$ reaches the max value at $i = 1$. So, the iterative ADP algorithm is not monotonically nonincreasing for $i = 0, 1, \dots$. While, for the iterative ADP algorithm (2.7)–(2.10), we have that the iterative ADP algorithm is monotonically nonincreasing for all $i = 0, 1, \dots$. Therefore, it is another apparent difference between the two iterative ADP algorithms.

From Theorem 2.2, we know that the iterative value function $V_i(x_k) \geq 0$ is a monotonically nonincreasing sequence with lower bound for iteration index $i = 1, 2, \dots$. Then, there exists a limit of the iterative value function $V_i(x_k)$. Define the iterative value function $V_\infty(x_k)$ as the limit of the iterative value function $V_i(x_k)$, i.e.,

$$V_\infty(x_k) = \lim_{i \rightarrow \infty} V_i(x_k). \quad (2.23)$$

Lemma 2.2 Let $\bar{v}_k^{N-1} = (\bar{v}_k, \bar{v}_{k+1}, \dots, \bar{v}_{N-1}) \in \mathfrak{A}_{x_k}^{N-k}$ be an arbitrary admissible control sequence. Define $\bar{\Phi}(x_k)$ as

$$\bar{\Phi}(x_k) = J(x_k, \bar{v}_k^{N-1}). \quad (2.24)$$

If we define a new value function $\bar{V}_i(x_k)$ as

$$\bar{V}_i(x_k) = \min_{u_k} \{U(x_k, u_k) + \bar{V}_{i-1}(x_{k+1})\}, \quad (2.25)$$

where $\bar{V}_0(x_k) = \bar{\Phi}(x_k)$, then for $i \rightarrow \infty$, we have

$$\bar{V}_\infty(x_k) = V_\infty(x_k), \quad (2.26)$$

where $V_\infty(x_k)$ is defined in (2.23).

Proof Let $\underline{\mu}_k = (\mu_k, \mu_{k+1}, \dots)$ be an arbitrary admissible control sequence. Define the value function $P_i(x_k)$ as (2.14). Then, we have

$$P_i(x_k) = \sum_{j=0}^{i-1} U(x_{k+j}, \mu_{k+j}) + \Phi(x_{k+i}). \quad (2.27)$$

Let $i \rightarrow \infty$ and we can get

$$P_\infty(x_k) = \sum_{j=0}^{\infty} U(x_{k+j}, \mu(x_{k+j})) + \lim_{i \rightarrow \infty} \Phi(x_{k+i}). \quad (2.28)$$

As $\underline{\mu}_k$ is an admissible control sequence, we have $x_{k+i} \rightarrow 0$ as $i \rightarrow \infty$. According to the definition of $\Phi(x_k)$ in (2.6), we can obtain

$$\lim_{i \rightarrow \infty} \Phi(x_{k+i}) = 0. \quad (2.29)$$

On the other hand, according to (2.11), let $i \rightarrow \infty$, we can get

$$V_\infty(x_k) = \lim_{i \rightarrow \infty} \left\{ \sum_{j=0}^{i-1} U(x_{k+j}, v_{i-j}(x_{k+j})) + \Phi(x_{k+i}) \right\}. \quad (2.30)$$

According to Lemma 2.1, we can obtain $V_\infty(x_k) \leq P_\infty(x_k)$. As $P_\infty(x_k)$ is finite, then according to (2.29) and (2.30), we have

$$V_\infty(x_k) = \lim_{i \rightarrow \infty} \left\{ \min_{\underline{u}_k^{k+i-1}} \left\{ \sum_{j=0}^{i-1} U(x_{k+j}, u_{k+j}) \right\} \right\}. \quad (2.31)$$

If we define $\bar{P}_i(x_k) = \sum_{j=0}^{i-1} U(x_{k+j}, \mu_{k+j}) + \bar{\Phi}(x_{k+i})$, then according to (2.28),

we have $\bar{P}_\infty(x_k) = \sum_{j=0}^{\infty} U(x_{k+j}, \mu(x_{k+j}))$, where $\lim_{i \rightarrow \infty} \bar{\Phi}(x_{k+i}) = 0$. As $\bar{V}_\infty(x_k) \leq \bar{P}_\infty(x_k)$, we can obtain

$$\begin{aligned}
\bar{V}_\infty(x_k) &= \lim_{i \rightarrow \infty} \left\{ \min_{\underline{u}_k^{k+i-1}} \left\{ \sum_{j=0}^{i-1} U(x_{k+j}, u_{k+j}) \right\} + \bar{\Phi}(x_{k+i}) \right\} \\
&= \lim_{i \rightarrow \infty} \left\{ \min_{\underline{u}_k^{k+i-1}} \left\{ \sum_{j=0}^{i-1} U(x_{k+j}, u_{k+j}) \right\} \right\} \\
&= V_\infty(x_k).
\end{aligned} \tag{2.32}$$

The proof is complete.

Next, we will prove that the iterative value function $V_i(x_k)$ converges to the optimal performance index function $J^*(x_k)$ as $i \rightarrow \infty$.

Theorem 2.3 *Let the iterative value function $V_i(x_k)$ be defined by (2.9). If the system state x_k is controllable, then the iterative value function $V_i(x_k)$ converges to the optimal performance index function $J^*(x_k)$ as $i \rightarrow \infty$, i.e.,*

$$V_i(x_k) \rightarrow J^*(x_k). \tag{2.33}$$

Proof According to the definition of $J^*(x_k)$ in (2.3), we have $J^*(x_k) \leq V_i(x_k)$. Let $i \rightarrow \infty$, and then we have

$$J^*(x_k) \leq V_\infty(x_k). \tag{2.34}$$

Next, as $P_q(x_k) - J^*(x_k) \geq 0$, taking $\underline{u}_k^{k+q-1} = \underline{u}_k^{*k+q-1}$ into $P_i(x_k)$ in (2.14), we can obtain $\Phi(x_{k+q}) - \sum_{j=q}^{N-1} U(x_{k+j}, u_{k+j}^*) \geq 0$, where N is the unspecified terminal time.

According to Lemma 2.2, we know that $\Phi(x_{k+q}) \rightarrow 0$ as $q \rightarrow \infty$. Let $\epsilon > 0$ be an arbitrary positive number. There exists a finite horizon admissible control sequence η_q such that

$$P_q(x_k) \leq J^*(x_k) + \epsilon. \tag{2.35}$$

On the other hand, according to Lemma 2.1, for any finite horizon admissible control η_q , we have

$$V_\infty(x_k) \leq V_q(x_k) \leq P_q(x_k). \tag{2.36}$$

Combining (2.35) and (2.36), we have $V_\infty(x_k) \leq J^*(x_k) + \epsilon$. As ϵ is an arbitrary positive number, we have

$$V_\infty(x_k) \leq J^*(x_k). \tag{2.37}$$

According to (2.34) and (2.37), we have

$$V_\infty(x_k) = J^*(x_k). \quad (2.38)$$

The proof is complete.

2.4 The ϵ -Optimal Control Algorithm

In the previous section, we proved that the iterative value function $V_i(x_k)$ converges to the optimal performance index function $J^*(x_k)$ as $i \rightarrow \infty$. This means that if we want to obtain the optimal performance index function $J^*(x_k)$, we should run the iterative ADP algorithm (2.7)–(2.10) for $i \rightarrow \infty$. But unfortunately, it is not achievable to run the algorithm for infinite number of times. For finite horizon optimal control, the infinite horizon ADP algorithm may not be effective. First, the infinite horizon optimal control makes the iterative value function converge to the optimum for $i \rightarrow \infty$, so the optimal control law is also convergent to the optimum. While for the finite horizon optimal control problem, for different initial state x_k , we should adopt different optimal control law. Second, the optimal control step for the infinite horizon optimal control is infinite. For the finite horizon control problems, for different initial state, the optimal step number is also different. Hence, to overcome this difficulty, a new ϵ -optimal control algorithm is established in this subsection.

2.4.1 The Derivation of the ϵ -Optimal Control Algorithm

In this subsection, we will introduce our method for iterative ADP with the consideration of the length of control sequences. For different x_k , we will use different i for the length of optimal control sequence. For a given error bound $\epsilon > 0$, the number i will be chosen so that the error between $J^*(x_k)$ and $V_i(x_k)$ is bounded with ϵ .

Theorem 2.4 *Let $\epsilon > 0$ be any small number and x_k be any controllable state. Let the iterative value function $V_i(x_k)$ be defined by (2.9) and $J^*(x_k)$ be the optimal performance index function. Then, there exists a finite i that satisfies*

$$|V_i(x_k) - J^*(x_k)| \leq \epsilon. \quad (2.39)$$

Definition 2.1 Let x_k be a controllable state vector. Let $\epsilon > 0$ be a small positive number. The approximate length of optimal control with respect to ϵ is defined as

$$K_\epsilon(x_k) = \min\{i : |V_i(x_k) - J^*(x_k)| \leq \epsilon\}. \quad (2.40)$$

Remark 2.3 An important property we must point out. For the iterative ADP algorithm (2.7)–(2.10), we have proven that for arbitrary initial value function $V_0(x_k) = \Phi(x_k)$, the iterative value function $V_i(x_k) \rightarrow J^*(x_k)$ as $i \rightarrow \infty$. For the finite horizon iterative ADP algorithm, the length $K_\epsilon(x_k)$ is different for different initial value function $\Phi(x_k)$. The following theorem will show this property.

Theorem 2.5 *Let $\Phi(x_k)$ and $\bar{\Phi}(x_k)$ are two different initial value functions. Let $V_i(x_k)$ be expressed by (2.9) and $\bar{V}_i(x_k)$ be expressed by (2.25). If define*

$$\bar{K}_\epsilon(x_k) = \min\{i : |\bar{V}_i(x_k) - J^*(x_k)| \leq \epsilon\}, \quad (2.41)$$

then we have $\bar{K}_\epsilon(x_k) \neq K_\epsilon(x_k)$.

Proof According to the Theorem 2.1 and the definition $K_\epsilon(x_k)$ in (2.40), there exists an $\tilde{\epsilon} \leq \epsilon$ that satisfies

$$\begin{aligned} V_i(x_k) &= \sum_{j=0}^{i-1} U(x_{k+j}, v_{i-j}(x_{k+j})) + \Phi(x_{k+i}) \\ &= J^*(x_k) + \tilde{\epsilon}. \end{aligned} \quad (2.42)$$

Taking the control sequence $\underline{v}_k^{k+i-1} = (v_i(x_k), v_{i-1}(x_{k+1}), \dots, v_0(x_{k+i-1}))$ into (2.25), we can get

$$\bar{V}_i(x_k) = \sum_{j=0}^{i-1} U(x_{k+j}, v_{i-j}(x_{k+j})) + \bar{\Phi}(x_{k+i}). \quad (2.43)$$

As $\underline{\bar{v}}_k^{N-1} = (\bar{v}_k, \bar{v}_{k+1}, \dots, \bar{v}_{N-1}) \in \mathfrak{A}_{x_k}^{N-k}$ is an arbitrary admissible control sequence, then there exists an control sequence $\underline{\tilde{v}}_k^{N-1}$ that makes $\bar{\Phi}(x_{k+i}) = J(x_k, \underline{\tilde{v}}_k^{N-1})$ satisfy

$$\begin{aligned} \bar{V}_i(x_k) &= \sum_{j=0}^{i-1} U(x_{k+j}, v_{i-j}(x_{k+j})) + \bar{\Phi}(x_{k+i}) \\ &\geq J^*(x_k) + \epsilon. \end{aligned} \quad (2.44)$$

According to the definitions of $K_\epsilon(x_k)$ and $\bar{K}_\epsilon(x_k)$, we can obtain $K_\epsilon(x_k) \leq \bar{K}_\epsilon(x_k)$. The proof is complete.

From Theorem 2.5 we can see that for different $\Phi(x_k)$, the approximate length of ϵ -optimal control $K_\epsilon(x_k)$ is also deferent. This makes it difficult to obtain the ϵ -optimal control law and the approximate length. Next, we will show that if we give a constraint for the initial value function $V_0(x_k)$, we can get that $K_\epsilon(x_k)$ is unique.

Theorem 2.6 Let $\underline{u}_{k+1}^{k+l*} = \{u_{k+1}^*, \dots, u_{k+l}^*\}$ be the optimal control sequence and $\Phi^*(x_{k+1}) = J^*(x_{k+1}, \underline{u}_{k+1}^{k+l*})$. If we let $V_0(x_{k+1}) = \Phi^*(x_{k+1})$, then we have

$$V_i(x_k) = J^*(x_k, \underline{u}_k^{k+l+i-1*}). \quad (2.45)$$

Proof According to Theorem 2.1, the iterative value function $V_i(x_k)$ can be expressed as (2.12). As $\Phi^*(x_{k+1}) = J^*(x_{k+1}, \underline{u}_{k+1}^{k+l-1*})$, we have

$$\Phi^*(x_{k+1}) = \min_{\underline{u}_{k+1}^{k+l}} \sum_{j=1}^l U(x_{k+j}, u_{k+j}). \quad (2.46)$$

Taking $\Phi^*(x_{k+1})$ into (2.12), we can obtain

$$\begin{aligned} V_i(x_k) &= \min_{\underline{u}_k^{k+l+i-1}} \left\{ \sum_{j=0}^{l+i-1} U(x_{k+j}, u_{k+j}) \right\} \\ &= J^*(x_k, \underline{u}_k^{k+l+i-1*}). \end{aligned} \quad (2.47)$$

The proof is complete.

From Theorem 2.6, we can see that if we can find an optimal control sequence $\underline{u}_{k+1}^{k+l*}$, then we can obtain the optimal control law and the optimal control length for the state x_k immediately. According to Theorem 2.3, we know that it requires to run the iterative ADP algorithm (2.7)–(2.10) for infinite times to obtain $J^*(x_{k+1})$ which is impossible to realize in the real world. Therefore, we give an ϵ -optimal control algorithm to obtain the approximate optimal performance index function and control law. Before the ϵ -optimal control iterative ADP algorithm, the following definition and lemma are necessary.

Definition 2.2 Let x_k be a controllable state vector and ϵ be a positive number. For $i = 1, 2, \dots$, define the set

$$\mathcal{T}_i^{(\epsilon)} = \{x_k \in \mathcal{T}_\infty : K_\epsilon(x_k) \leq i\}. \quad (2.48)$$

When $x_k \in \mathcal{T}_i^{(\epsilon)}$, to find the optimal control sequence which means iterative value function less than or equal to $J^*(x_k) + \epsilon$, we only need to consider the control sequences $\underline{u}_k$ with length $|\underline{u}_k| \leq i$. The set $\mathcal{T}_i^{(\epsilon)}$ has the following properties.

Lemma 2.3 [31] Let $\epsilon > 0$ and $i = 1, 2, \dots$. Then,

- (i) $x_k \in \mathcal{T}_i^{(\epsilon)}$ if and only if $V_i(x_k) \leq J^*(x_k) + \epsilon$;
- (ii) $\mathcal{T}_i^{(\epsilon)} \subseteq \mathcal{T}_i$;
- (iii) $\mathcal{T}_i^{(\epsilon)} \subseteq \mathcal{T}_{i+1}^{(\epsilon)}$;
- (iv) $\cup_i \mathcal{T}_i^{(\epsilon)} = \mathcal{T}_\infty$;
- (v) If $\epsilon > \delta > 0$, then $\mathcal{T}_i^{(\epsilon)} \supseteq \mathcal{T}_i^{(\delta)}$.

Next, we will show the ϵ -optimal control iterative ADP algorithm. First, let $\widehat{\underline{u}}_0^{K-1} = (u_0, u_1, \dots, u_{K-1})$ is arbitrary finite horizon admissible control sequence and the corresponding state sequence is $\widehat{\underline{x}}_0^K = (x_0, x_1, \dots, x_K)$ where $x_K = 0$. We can see that the initial control sequence $\widehat{\underline{u}}_0^{K-1} = (u_0, u_1, \dots, u_{K-1})$ may not be optimal which means the initial control step number K may not be optimal and also the law of the initial control sequence $\widehat{\underline{u}}_0^{K-1}$ may not be optimal. In the following part, we will show that the control step number and the control law are both optimized in iterative ADP algorithm simultaneously.

For the state x_{K-1} , we have $F(x_{K-1}, u_{K-1}) = 0$. Then we run the iterative ADP algorithm proposed in [24, 25, 31] at x_{K-1} until

$$|V_{l_1}(x_{K-1}) - J^*(x_{K-1})| \leq \epsilon \quad (2.49)$$

holds where $l_1 > 0$ is a positive integer number. This means $x_{K-1} \in \mathcal{T}_{l_1}^{(\epsilon)}$ and the optimal control step number $K_\epsilon(x_{K-1}) = l_1$.

Then, considering x_{K-j} , $j = 0, 1, \dots, K$, we have $F(x_{K-j}, u_{K-j}) = x_{K-j+1}$. For x_{K-j} , if

$$|V_{l_{j-1}}(x_{K-j}) - J^*(x_{K-j})| \leq \epsilon \quad (2.50)$$

holds, then we say $x_{K-j} \in \mathcal{T}_{l_{j-1}}^{(\epsilon)}$, and $u_{l_{j-1}}(x_{K-j})$ is the corresponding ϵ -optimal control law. If not, $x_{K-j} \notin \mathcal{T}_{l_{j-1}}^{(\epsilon)}$ and then we run the iterative ADP algorithm as

$$v_{l_{j-1}}(x_{K-j}) = \arg \min_{u_{K-j}} \{U(x_{K-j}, u_{K-j}) + V_{l_{j-1}}(x_{K-j+1})\} \quad (2.51)$$

and

$$V_{l_{j-1}+1}(x_{K-j}) = U(x_{K-j}, u_{l_{j-1}}(x_{K-j})) + V_{l_{j-1}}(F(x_{K-j}, v_{l_{j-1}}(x_{K-j}))). \quad (2.52)$$

For $i = 1, 2, \dots$, the iterative ADP algorithm between

$$v_{l_{j-1}+i}(x_{K-j}) = \arg \min_{u_{K-j}} \{U(x_{K-j}, u_{K-j}) + V_{l_{j-1}+i}(x_{K-j+1})\} \quad (2.53)$$

and

$$V_{l_{j-1}+i+1}(x_{K-j}) = U(x_{K-j}, u_{l_{j-1}+i}(x_{K-j})) + V_{l_{j-1}+i}(F(x_{K-j}, v_{l_{j-1}+i}(x_{K-j}))) \quad (2.54)$$

until the following inequality

$$|V_{l_j}(x_{K-j}) - J^*(x_{K-j})| \leq \epsilon \quad (2.55)$$

holds where $l_j > 0$ is a positive integer number. So we can obtain $x_{K-j} \in \mathcal{T}_{l_j}^{(\epsilon)}$ and the optimal control step number $K_\epsilon(x_{K-j}) = l_j$.

2.4.2 Properties of the ϵ -Optimal Control Algorithm

We can see that an error ϵ between $J^*(x_k)$ and $V_i(x_k)$ is introduced into the iterative ADP algorithm which makes the iterative value function $V_i(x_k)$ converge within finite iteration step i . In this subsection, we will show that the corresponding control is also an effective control that makes the iterative value function reach the optimal within error bound ϵ . According to Lemma 2.3, we have the following theorem.

Theorem 2.7 *Let $\epsilon > 0$ and $i = 0, 1, 2, \dots$. If $x_k \in \mathcal{T}_i^{(\epsilon)}$, then for any $x'_k \in \mathcal{T}_i^{(\epsilon)}$, we have the following inequality:*

$$|V_i(x'_k) - J^*(x'_k)| \leq \epsilon. \quad (2.56)$$

Proof The theorem can be easily proven by contradiction. Assume that the conclusion is false. Then for some $x'_k \in \mathcal{T}_i^{(\epsilon)}$, we have

$$V_i(x'_k) > J^*(x'_k) + \epsilon. \quad (2.57)$$

So we can get

$$\begin{aligned} K_\epsilon(x'_k) &= \min\{j : |V_j(x'_k) - J^*(x'_k)| \leq \epsilon\} \\ &> i. \end{aligned} \quad (2.58)$$

Then, according to Definition 2.2, we can obtain $x'_k \notin \mathcal{T}_i^{(\epsilon)}$ which is contradiction with the assumption of $x'_k \in \mathcal{T}_i^{(\epsilon)}$. So the conclusion holds.

Corollary 2.1 *For $i = 0, 1, \dots$, let $\mu_\epsilon^*(x_k)$ be expressed as*

$$\mu_\epsilon^*(x_k) = \arg \min_{u_k} \{U(x_k, u_k) + V_i(F(x_k, u_k))\} \quad (2.59)$$

that makes the iterative value function (2.39) hold for $x_k \in \mathcal{T}_i^{(\epsilon)}$. Then for $x'_k \in \mathcal{T}_i^{(\epsilon)}$, $\mu_\epsilon^(x'_k)$ satisfies*

$$|V_i(x'_k) - J^*(x'_k)| \leq \epsilon. \quad (2.60)$$

Then, we have the following Theorem.

Theorem 2.8 *For $i = 0, 1, \dots$, if we let $x_k \in \mathcal{T}_{i+1}^{(\epsilon)}$ and $\mu_\epsilon^*(x_k)$ be expressed in (2.59), then $F(x_k, \mu_\epsilon^*(x_k)) \in \mathcal{T}_i^{(\epsilon)}$. In other words, if $K_\epsilon(x_k) = i + 1$, then $K_\epsilon(F(x_k, \mu_\epsilon^*(x_k))) \leq i$.*

Proof Since $x_k \in \mathcal{T}_{i+1}^{(\epsilon)}$, by Lemma 2.3(i) we know that

$$V_{i+1}(x_k) \leq J^*(x_k) + \epsilon. \quad (2.61)$$

According to the expression of $\mu_\epsilon^*(x_k)$ in (2.59), we have

$$V_{i+1}(x_k) = U(x_k, \mu_\epsilon^*(x_k)) + V_i(F(x_k, \mu_\epsilon^*(x_k))). \quad (2.62)$$

From (2.61) and (2.62), we have

$$\begin{aligned} V_i(F(x_k, \mu_\epsilon^*(x_k))) &= V_{i+1}(x_k) - U(x_k, \mu_\epsilon^*(x_k)) \\ &\leq J^*(x_k) + \epsilon - U(x_k, \mu_\epsilon^*(x_k)). \end{aligned} \quad (2.63)$$

On the other hand, we have

$$J^*(x_k) \leq U(x_k, \mu_\epsilon^*(x_k)) + J^*(F(x_k, \mu_\epsilon^*(x_k))). \quad (2.64)$$

Putting (2.64) into (2.63) we obtain

$$V_i(F(x_k, \mu_\epsilon^*(x_k))) \leq J^*(F(x_k, \mu_\epsilon^*(x_k))) + \epsilon. \quad (2.65)$$

By Lemma 2.3, we have

$$F(x_k, \mu_\epsilon^*(x_k)) \in \mathcal{T}_i^{(\epsilon)}. \quad (2.66)$$

So if $K_\epsilon(x_k) = i + 1$, we know that $x_k \in \mathcal{T}_{i+1}^{(\epsilon)}$ and $F(x_k, \mu_\epsilon^*(x_k)) \in \mathcal{T}_i^{(\epsilon)}$ according to (2.66). Therefore, we have

$$K_\epsilon(F(x_k, \mu_\epsilon^*(x_k))) \leq i. \quad (2.67)$$

The theorem is proven.

Corollary 2.2 For $i = 0, 1, \dots$, let $\mu_\epsilon^*(x_k)$ be expressed in (2.59) where the iterative value function $|V_{i+1}(x_k) - J^*(x_k)| \leq \epsilon$. Then for any $x'_k \in \mathcal{T}_i^{(\epsilon)}$, we have the following inequality:

$$|V_i(x'_k) - J^*(x'_k)| \leq \epsilon. \quad (2.68)$$

Now we look back to the optimal control problem with respect to performance index function. If the initial state x_0 is fixed, we will show that if we choose x_0 to run the iterative value function we can obtain the ϵ -optimal control.

Theorem 2.9 Let x_0 be the fixed initial state, $\mu_\epsilon^*(x_0)$ satisfies (2.59) at $k = 0$. If x_k , $k = 0, 1, \dots, N$, is the state under the control law $\mu_\epsilon^*(x_k)$, then we have $|V_i(x_k) - J^*(x_k)| \leq \epsilon$ for any k .

Proof For the system (2.1) with respect to the performance index function (2.2), we have $x_0 \in \mathcal{T}_N^{(\epsilon)}$ and $K_\epsilon(x_0) = N$. Then for small ϵ , there exists an ϵ -optimal control sequence

$$\underline{\mu}_\epsilon^*(x_0) = (\mu_\epsilon^*(x_0), \mu_\epsilon^*(x_1), \dots, \mu_\epsilon^*(x_{N-1})), \quad (2.69)$$

which stabilizes the system (2.1) within finite time N and minimizes the performance index function (2.2). Then obviously, we have $x_N \in \mathcal{T}_0$, $x_{N-1} \in \mathcal{T}_1^{(\epsilon)}$, $\dots$, $x_0 \in \mathcal{T}_N^{(\epsilon)}$ where $0 = \mathcal{T}_0 \subseteq \mathcal{T}_1^{(\epsilon)} \subseteq \dots \subseteq \mathcal{T}_N^{(\epsilon)}$. So according to Theorem 2.8 and Corollary 2.2, we have that the ϵ -optimal control law μ_ϵ^* obtained by the initial state x_0 using the iterative ADP algorithm is effective for the states $x_1, x_2, \dots, x_N$. The proof is complete.

We can see that if we choose x_0 to run the iterative value function we can obtain the ϵ -optimal control. While if the initial state x_0 is unfixed, then we do not know which one should be used to implement the iterative ADP algorithm. In the next section, we will solve this problem.

2.4.3 The ϵ -Optimal Control Algorithm for Unfixed Initial State

For a lot of practical nonlinear systems, the initial state x_0 cannot be fixed. Instead, the initial state belongs to a domain and we define the domain of initial state as Ω_0 where $\Omega_0 \subseteq \mathbb{R}^n$. Then, we have $x_0 \in \Omega_0$. For this case, if we only choose one state $x_0^{(i)} \in \Omega_0$ to run the iterative ADP algorithm and get corresponding ϵ -optimal control μ_ϵ^* , then the ϵ -optimal control μ_ϵ^* may not be ϵ -optimal for all $x_0 \in \Omega_0$ because there may exist a state $x_0^{(j)} \in \Omega_0$ such that $x_0^{(i)} \in \mathcal{T}_i^{(\epsilon)}$ while $x_0^{(j)} \in \mathcal{T}_j^{(\epsilon)} \setminus \mathcal{T}_i^{(\epsilon)}$ where $j > i$. If we let

$$I = \max \left\{ i : x_0 \in \mathcal{T}_i^{(\epsilon)} \text{ s.t. } x_0 \in \Omega_0 \right\}, \quad (2.70)$$

then according to Corollary 2.2, we should find the initial state $x_0 \in \mathcal{T}_I^{(\epsilon)}$ to obtain the most effective ϵ -optimal control. Thus, the next job is to obtain the state $x_0 \in \mathcal{T}_I^{(\epsilon)}$. For this case, there are two methods which are “entire state space searching method” and “partial state space searching method” to obtain the ϵ -optimal control $\mu_\epsilon^*(x_k)$ for $k = 0, 1, \dots$

(1) *Entire state space searching method.*

Choosing randomly an array of enough states

$$X = (x^{(1)}, x^{(2)}, \dots, x^{(\mathcal{Q})}) \quad (2.71)$$

from the entire initial state space Ω , where $\mathcal{Q} > 0$ is a positive integer number. First, we solve (2.7) where $x_k = x^{(1)}, x^{(2)}, \dots, x^{(\mathcal{Q})}$, respectively and $V_0(x_{k+1})$ is the converged iterative performance index function obtained by (2.49)–(2.55) at x_{k+1} . If for $0 \leq j_1 \leq \mathcal{Q}$ and $x_k = x^{(j_1)} \in X$, the inequality

$$|V_1(x^{(j_1)}) - J^*(x^{(j_1)})| \leq \epsilon \quad (2.72)$$

holds, then we have $x^{(j_1)} \in \mathcal{T}_1^{(\epsilon)}$. We record the iterative value function V_1 and let

$$X_1 = \{x^{(j_1)} \in X : |V_1(x^{(j_1)}) - J^*(x^{(j_1)})| \leq \epsilon\}. \quad (2.73)$$

So we can repeat the process (2.72)–(2.73) for iteration index $i = 1, 2, \dots$ to solve (2.9), where $x_k \in X \setminus \{X_1 \cup X_2 \cup \dots \cup X_{i-1}\}$. If for $0 \leq j_i \leq \mathcal{Q}$ and $x_k = x^{(j_i)} \in X \setminus \{X_1 \cup X_2 \cup \dots \cup X_{i-1}\}$, the inequality

$$|V_i(x^{(j_i)}) - J^*(x^{(j_i)})| \leq \epsilon \quad (2.74)$$

holds, then we have $x^{(j_i)} \in \mathcal{T}_i^{(\epsilon)}$. We record the iterative value function V_i and let

$$X_i = \{x^{(j_i)} \in X \setminus \{X_1 \cup X_2 \cup \dots \cup X_{i-1}\} : |V_i(x^{(j_i)}) - J^*(x^{(j_i)})| \leq \epsilon\}. \quad (2.75)$$

For the initial state x_0 , if $|V_i(x_0) - J^*(x_0)| \leq \epsilon$ holds, then the ϵ -optimal performance index function is obtained and the corresponding control law is the ϵ -optimal control law μ_ϵ^* .

Remark 2.4 The structure of the entire state space searching method is clear and simple which is based on the idea of dynamic programming. This is the merit of the entire space searching method. But it also possesses serious shortcomings. First, the array of states X in (2.71) should include enough state points which is distributed for the entire initial state space Ω . Second, for each state point $x^{(j_i)} \in X$, the iterative algorithm (2.71)–(2.74) should run one time and then record X_i in (2.75). So the computation capacity is very huge. Especially for neural network implement, it means that the neural network should be trained at every state point for the entire state space to obtain the optimal control and the “curse of dimensionality” cannot avoid. Therefore, the entire state space searching method is very difficult to apply to the optimal control problem of real-world systems.

(2) Partial state space searching method

In the partial state space searching method, it is not necessary to search the entire state space to obtain the optimal control. Instead, only the boundary of the domain of initial state Ω is searched to obtain the ϵ -optimal control which overcomes the difficulty of the “curse of dimensionality” effectively.

Theorem 2.10 *Let $\Omega_0 \subseteq \mathbb{R}^n$ be the domain of initial state and the initial state $x_0 \in \Omega_0$. If Ω_0 is a convex set on $\mathbb{R}^n$, then $x_0^{(I)}$ is a boundary point of Ω_0 where I is defined in (2.70).*

Proof The theorem can be proven by contradiction. Assume that $x_0^{(I)}$ is a interior point of Ω_0 . Without loss of generality, let the point be $x_a = x_0^{(I)}$. Make a beeline between the origin and x_a . Let the point of intersection between the beeline and the boundary of the domain Ω_0 be x_b . Let the point of intersection between the extended line and the boundary of the domain Ω_0 be x_c . As $x_0^{(I)}$ is an interior point of Ω_0 which is convex, according to the property of convex set, for all $x_0^{(j)} \in \Omega_0$, $j = 0, 1, \dots$, there exists a positive real number $0 \leq \lambda \leq 1$ that makes

$$x_0^{(j_a)} = \lambda x_0^{(j_b)} + (1 - \lambda) x_0^{(j_c)} \quad (2.76)$$

hold, where j_a , j_b and j_c are nonnegative integer numbers.

If let $x_a = x_0^{(I)} = x_0^{(j_a)}$, $x_b = x_0^{(j_b)}$ and $x_c = x_0^{(j_c)}$, then we have

$$x_a = \lambda x_b + (1 - \lambda) x_c. \quad (2.77)$$

If we assume that $x_a \in \mathcal{T}_a^{(\epsilon)}$, $x_b \in \mathcal{T}_b^{(\epsilon)}$ and $x_c \in \mathcal{T}_c^{(\epsilon)}$, then we have

$$\mathcal{T}_c^{(\epsilon)} \subseteq \mathcal{T}_a^{(\epsilon)} \quad (2.78)$$

because $x_a = x_0^{(I)}$ where I is expressed in (2.70). Then we can obtain

$$\begin{aligned} I &= K_\epsilon(F(x_a, \mu_\epsilon^*(x_a))) \\ &\geq K_\epsilon(F(x_c, \mu_\epsilon^*(x_c))) \\ &= c. \end{aligned} \quad (2.79)$$

While on the other hand, as x_c is the point of intersection between the extended beeline and the boundary of Ω_0 , obviously the point x_c is farther from the origin. So we have

$$\begin{aligned} K_\epsilon(F(x_a, \mu_\epsilon^*(x_a))) &\leq K_\epsilon(F(x_c, \mu_\epsilon^*(x_c))) \\ &= c, \end{aligned} \quad (2.80)$$

which is a contradiction to (2.79). So $x_0^{(I)}$ cannot be expressed as (2.76). Then the assumption is false and therefore $x_0^{(I)}$ must be the boundary point of Ω_0 .

Remark 2.5 Theorem 2.10 gives an important property obtaining the optimal control law. It means that if the domain of initial state Ω_0 is convex, it is not necessary to search all the state points in Ω_0 . Instead, it only requires to search the boundary of Ω_0 and therefore the computational burden is much reduced.

2.4.4 The Expressions of the ϵ -Optimal Control Algorithm

In [31], we analyzed the ϵ -optimal control iterative ADP algorithm when the initial state is fixed. In [24, 25], we give an iterative ADP algorithm for unfixed initial state while it requires the control system can reach to zero directly. In this chapter, we develop a new ϵ -optimal control iterative ADP algorithm for unfixed initial state, while the strict initial condition in [24, 25], can be omitted. To sum up, the finite horizon ϵ -optimal control problem with finite time can be separated into four cases.

Case 1. The initial state x_0 is fixed and for any state $x_k \in \mathbb{R}^n$, there exists a control $u_k \in \mathbb{R}^m$ that stabilizes the state to zero directly (proposed in [31]).

Case 2. The initial state $x_0 \in \Omega_o$ is unfixed and for any state $x_k \in \mathbb{R}^n$, there exists a control $u_k \in \mathbb{R}^m$ that stabilizes the state to zero directly (proposed in [24, 25]).

Case 3. The initial state x_0 is fixed and $\exists x_k \in \mathbb{R}^n$ such that $F(x_k, u_k) = 0$ is no solution for all $u_k \in \mathbb{R}^m$ (proposed in [31]).

Case 4. The initial state $x_0 \in \Omega_o$ is unfixed and $\exists x_k \in \mathbb{R}^n$ such that $F(x_k, u_k) = 0$ is no solution for all $u_k \in \mathbb{R}^m$ (developed in this chapter).

We can see that Case 1– Case 3 are the special cases of Case 4. Therefore, we can say that the present iterative ADP algorithm is the most effective one. Given the preparations, we now summarize the iterative ADP algorithms as follows:

Step 01. Give the initial state space Ω_0 , the max iterative number $i_{\max}$ and the computation precision ϵ .

Step 02. Let $\bar{\Omega}_o$ be the boundary of the domain of initial state Ω_o . Grid $\bar{\Omega}_o$ into $\bar{P}$ subsets which are expressed as $\bar{\Omega}_o^{(1)}, \bar{\Omega}_o^{(2)}, \dots, \bar{\Omega}_o^{(\bar{P})}$, where $\bar{\Omega}_o = \cup_{j=1}^{\bar{P}} \bar{\Omega}_o^{(j)}$ and $\bar{P} > 0$ is a positive integer number. For $j = 1, 2, \dots, \bar{P}$, let X_0 be expressed as $X_0 = (x^{(1)}, \dots, x^{(\bar{P})})$, and then $x_0^{(j)}$ satisfies $x_0^{(j)} \in \bar{\Omega}_o^{(j)}$.

Step 03. For $j = 1, 2, \dots, \bar{P}$, let $x_0 = x_0^{(j)}$ and loop (2.49)–(2.55).

Step 04. For $x_0 = x_0^{(j)}$, obtain $x_0^{(j)} \in \mathcal{T}_{i_j}^{(\epsilon)}$ record the iterative value function $V_{i_j}(x_0^{(j)})$, and the control law $\mu_\epsilon^*(x_0^{(j)})$.

Step 05. Let I be expressed as (2.70), we get $x_0^{(\bar{j})} \in \mathcal{T}_I^{(\epsilon)}$ and $K_\epsilon(x_0^{(\bar{j})}) = I$.

Step 06. Record the corresponding iterative value function $V_I(x_0^{(\bar{j})})$, and the control law $\mu_\epsilon^*(x_0^{(\bar{j})})$.

Step 07. Stop.

2.5 Neural Network Implementation for the ϵ -Optimal Control Scheme

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 NN is represented by

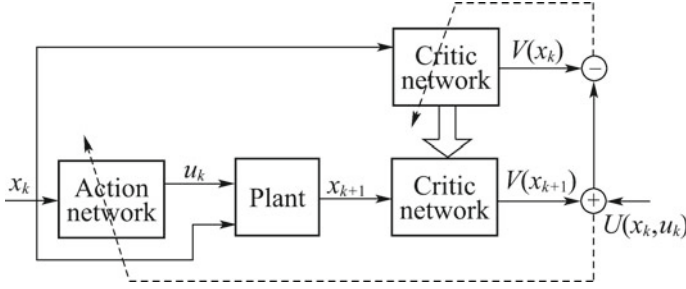


Fig. 2.1 The structure diagram of the algorithm (From [26] Fig. 2.)

$$\hat{F}(X, Y, W) = W^T \sigma(Y^T X), \quad (2.81)$$

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

The NN estimation error can be expressed by

$$F(X) = F(X, Y^*, W^*) + \varepsilon(X), \quad (2.82)$$

where Y^* , W^* are the ideal weight parameters, and $\varepsilon(X)$ is the reconstruction error.

Here, there are two networks, which are critic network and action network, respectively. Both neural networks are chosen as three-layer feedforward network. The whole structure diagram is shown in Fig. 2.1.

2.5.1 The Critic Network

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

$$\hat{V}_i(x_k) = W_{ci}^T \sigma(V_{ci}^T x_k). \quad (2.83)$$

The target function can be written as

$$V_{i+1}(x_k) = x_k^T Q x_k + \hat{u}_i^T(x_k) R \hat{u}_i(x_k) + \hat{V}_i(x_{k+1}). \quad (2.84)$$

Then we define the error function for the critic network as

$$e_{ci}(k) = \hat{V}_{i+1}(x_k) - V_{i+1}(x_k). \quad (2.85)$$

The objective function to be minimized in the critic network is

$$E_{ci}(k) = \frac{1}{2} e_{ci}^2(k). \quad (2.86)$$

So the gradient-based weight update rule for the critic network [18, 24] is given by

$$w_{c(i+1)}(k) = w_{ci}(k) + \Delta w_{ci}(k), \quad (2.87)$$

$$\Delta w_{ci}(k) = \alpha_c \left[-\frac{\partial E_{ci}(k)}{\partial w_{ci}(k)} \right], \quad (2.88)$$

$$\frac{\partial E_{ci}(k)}{\partial w_{ci}(k)} = \frac{\partial E_{ci}(k)}{\partial \hat{V}_i(x_k)} \frac{\partial \hat{V}_i(x_k)}{\partial w_{ci}(k)}, \quad (2.89)$$

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

2.5.2 The Action Network

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

$$\hat{v}_i(x_k) = W_{ai}^T \sigma(V_{ai}^T x_k). \quad (2.90)$$

The target of the output of the action network is given by (2.53) for $i = 1, 2, \dots$. So we can define the output error of the action network as

$$e_{ai}(k) = \hat{v}_i(x_k) - v_i(x_k). \quad (2.91)$$

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

$$E_{ai}(k) = \frac{1}{2} e_{ai}^2(k). \quad (2.92)$$

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

$$w_{a(i+1)}(k) = w_{ai}(k) + \Delta w_{ai}(k), \quad (2.93)$$

$$\Delta w_{ai}(k) = \beta_a \left[-\frac{\partial E_{ai}(k)}{\partial w_{ai}(k)} \right], \quad (2.94)$$

$$\frac{\partial E_{ai}(k)}{\partial w_{ai}(k)} = \frac{\partial E_{ai}(k)}{\partial e_{ai}(k)} \frac{\partial e_{ai}(k)}{\partial v_i(k)} \frac{\partial v_i(k)}{\partial w_{ai}(k)}, \quad (2.95)$$

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

2.6 Simulation Study

To evaluate the performance of our iterative ADP algorithm, we give an example with quadratic utility functions for numerical experiment. Our example is also used in [24, 25, 31]. We consider the system

$$\begin{aligned} x_{k+1} &= F(x_k, u_k) \\ &= x_k + \sin(0.1x_k^2 + u_k), \end{aligned} \quad (2.96)$$

where $x_k, u_k \in \mathbb{R}$, and $k = 0, 1, 2, \dots$. The domain of initial state is expressed as

$$\Omega_0 = \{x_0 | 0.8 \leq x_0 \leq 1.5\}. \quad (2.97)$$

The performance index function is quadratic form with finite-time horizon that is expressed as (2.2) with $U(x_k, u_k) = x_k^T Q x_k + u_k^T R u_k$, where the matrix $Q = R = I$ and I denotes the identity matrix with suitable dimensions.

We can see that for the initial state $0.8 \leq x_0 \leq 1$, there exists a control $u_0 \in \mathbb{R}$ that makes $x_1 = F(x_0, u_0) = 0$. Thus the situation is then belongs to Case 2. While for the initial state $1 < x_0 \leq 1.5$, there does not exist a control $u_0 \in \mathbb{R}$ that makes $x_1 = F(x_0, u_0) = 0$. Thus the situation then belongs to Case 4. Then we will compute the ϵ -optimal control law for $0.8 \leq x_0 \leq 1$ and $1 < x_0 \leq 1.5$ respectively. The computation error of the iterative ADP is given as $\epsilon = 10^{-6}$. The neural network

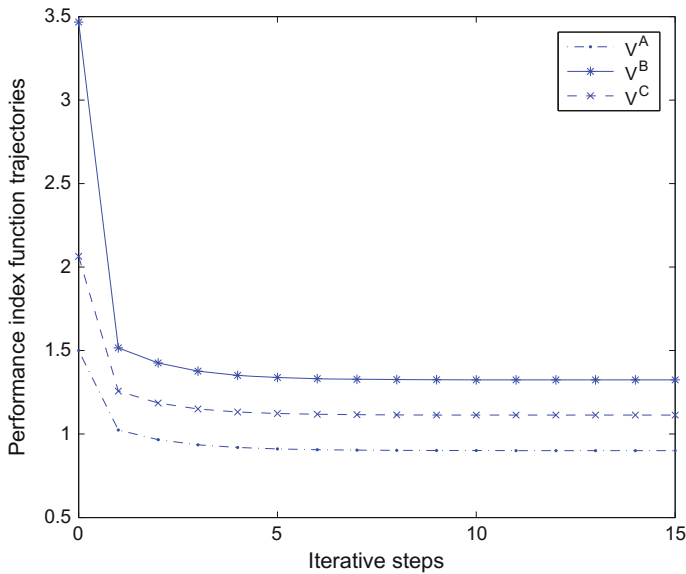


Fig. 2.2 The convergence of the iterative value functions: V^A , V^B and V^C (From [26] Fig. 3.)

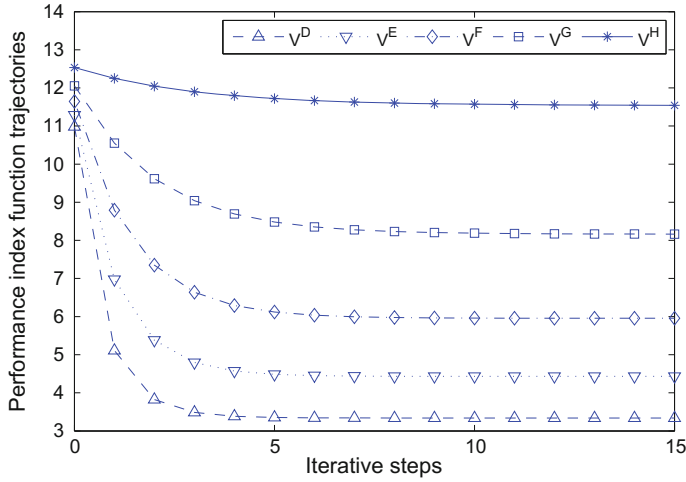


Fig. 2.3 The convergence of the iterative value functions: V^D , V^E , V^F , V^G and V^H (From [26] Fig. 4.)

structure of the algorithm is shown in Fig. 2.1. The critic network and the action network are chosen as three-layer BP neural networks with the structure 2-8-1 and 2-8-1 respectively. For $0.8 \leq x_0 \leq 1$, we run the iterative ADP algorithm for Case 2. The search step is 0.1 from $x_k = 0.8$ to $x_k = 1$. We illustrate the convergence of the iterative value functions at 3 points which are $x_A = 0.8$, $x_B = 0.9$ and $x_C = 1$. The corresponding convergence trajectories are V^A , V^B and V^C which are showed in Fig. 2.2, respectively.

For $1 < x_0 \leq 1.5$, we run the iterative ADP algorithm for Case 4. The search step is 0.1 from $x_k = 1$ to $x_k = 1.5$. There are 5 state points which are $x_D = 1.1$, $x_E = 1.2$, $x_F = 1.3$, $x_G = 1.4$, and $x_H = 1.5$. For each state point, we should give a finite horizon admissible control sequence as the initial control sequence. For convenience, the length of all the initial control sequence is 2. The control sequences are ${}_D\hat{u}_0^1 = (-\sin^{-1}(0.3) - 0.121, -\sin^{-1}(0.8) - 0.064)$, ${}_E\hat{u}_0^1 = (-\sin^{-1}(0.4) - 0.144, -\sin^{-1}(0.8) - 0.064)$, ${}_F\hat{u}_0^1 = (-\sin^{-1}(0.5) - 0.169, -\sin^{-1}(0.8))$, ${}_G\hat{u}_0^1 = (-\sin^{-1}(0.6) - 0.196, -\sin^{-1}(0.8))$ and ${}_H\hat{u}_0^1 = (-\sin^{-1}(0.7) - 0.225, -\sin^{-1}(0.8))$. The corresponding state trajectories are ${}_D\hat{x}_0^2 = (1.1, 0.8, 0)$, ${}_E\hat{x}_0^2 = (1.2, 0.8, 0)$, ${}_F\hat{x}_0^2 = (1.3, 0.8, 0)$, ${}_G\hat{x}_0^2 = (1.4, 0.8, 0)$, ${}_H\hat{x}_0^2 = (1.5, 0.8, 0)$.

We run the iterative ADP algorithm for Case 4 at state points x_D , x_E , x_F , x_G and x_H . For each iterative step, the critic network and the action network are also trained for 1000 steps under the learning rate $\alpha = 0.05$ so that the given neural network accuracy $\varepsilon = 10^{-8}$ is reached. After 15 iterative steps, the corresponding convergent trajectories of the iterative value functions are V^D , V^E , V^F , V^G and V^H which are showed in Fig. 2.3.

From the simulation results we have $x_A \in \mathcal{T}_5^{(\epsilon)}$, $x_B \in \mathcal{T}_5^{(\epsilon)}$, $x_C \in \mathcal{T}_6^{(\epsilon)}$, $x_D \in \mathcal{T}_6^{(\epsilon)}$, $x_E \in \mathcal{T}_6^{(\epsilon)}$, $x_F \in \mathcal{T}_6^{(\epsilon)}$, $x_G \in \mathcal{T}_7^{(\epsilon)}$ and $x_H \in \mathcal{T}_7^{(\epsilon)}$ and we have $I = 7$. To show the effectiveness of the optimal control, we arbitrarily choose 3 state points in Ω_0 such as $x_\alpha = 0.8$, $x_\beta = 1$ and $x_\gamma = 1.5$. Applying the optimal control law of $\mu_\epsilon^*(x_H)$ to the 3 state points, we obtain the following results exhibited Figs. 2.4 and 2.5.

2.7 Conclusions

In this chapter, we developed an effective iterative ADP algorithm for finite horizon ϵ -optimal control of discrete-time nonlinear systems with unfixed initial state. The iterative ADP algorithm can be implemented by an arbitrary admissible control sequence while the initial constraint which requires the system reach to zero directly is omitted. Convergence of the iterative value function for the iterative ADP algorithm is proven and the ϵ -optimal number of control steps can also be obtained. Neural networks are used to implement the iterative ADP algorithm. Finally, a simulation example is given to illustrate the performance of the present algorithm.

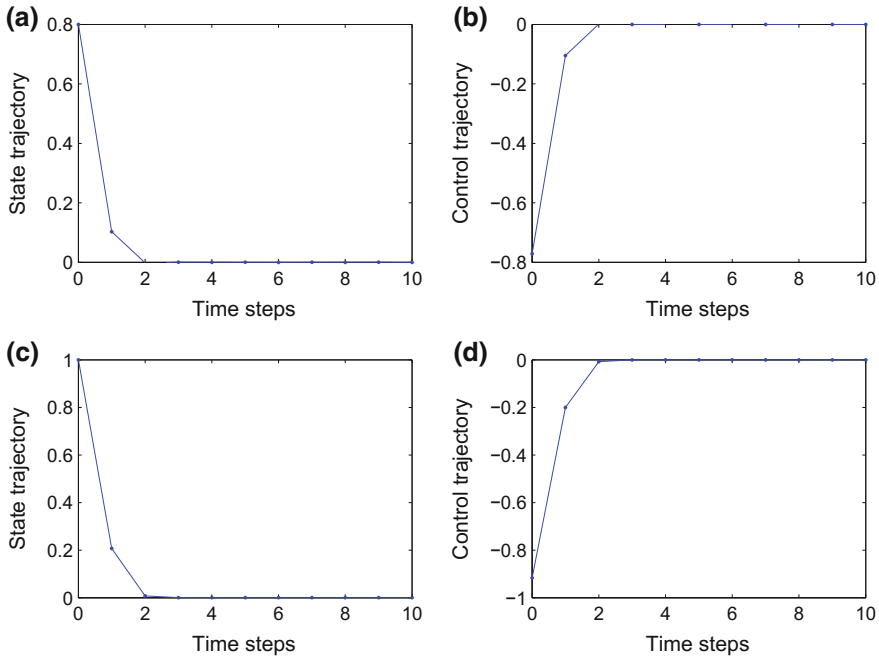


Fig. 2.4 Simulation results. (a) State trajectory for $x_\alpha = 0.8$. (b) Control trajectory for $x_\alpha = 0.8$. (c) State trajectory for $x_\beta = 1$. (d) Control trajectory for $x_\beta = 1$. (From [26] Fig. 5.)

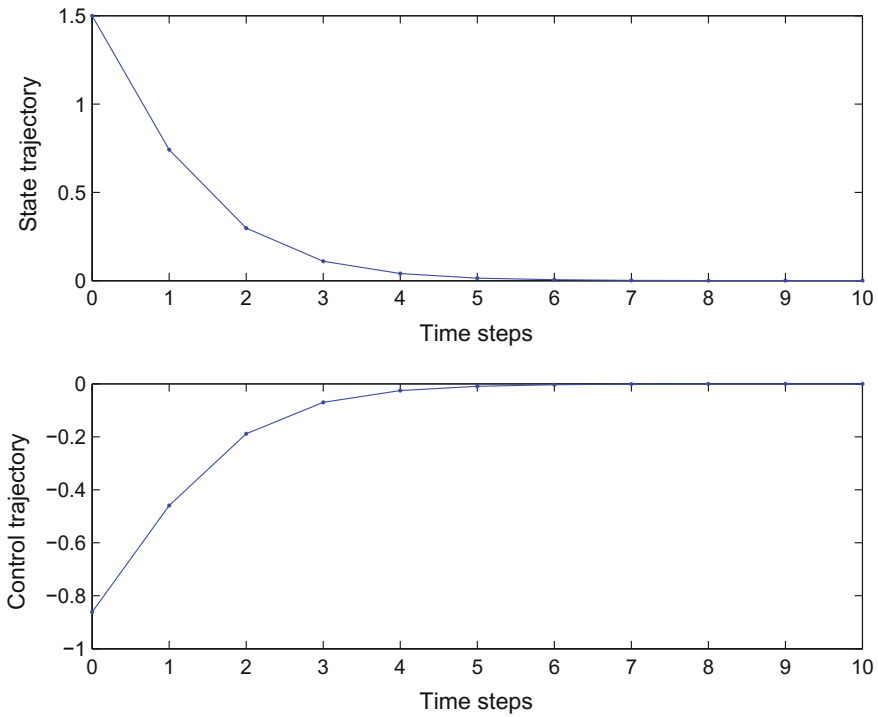


Fig. 2.5 Simulation results for the state $x_\gamma = 1.5$. **(a)** State trajectories. **(b)** Control trajectories. (From [26] Fig. 6.)

References

1. Al-Tamimi, A., Abu-Khalaf, M., Lewis, F.L.: Adaptive critic designs for discrete-time zero-sum games with application to H_∞ control. *IEEE Trans. Syst. Man. Cybern. Part B Cybern.* **37**(1), 240–247 (2008)
2. Bellman, R.E.: *Dynamic Programming*. Princeton University Press, Princeton (1957)
3. Dierks, T., Thumati, B.T., Jagannathan, S.: Optimal control of unknown affine nonlinear discrete-time systems using offline-trained neural networks with proof of convergence. *Neural Netw.* **22**(5–6), 851–860 (2009)
4. Ichihara, H.: Optimal control for polynomial systems using matrix sum of squares relaxations. *IEEE Trans. Autom. Control* **54**(5), 1048–1053 (2009)
5. Kioskeridis, I., Mademlis, C.: A unified approach for four-quadrant optimal controlled switched reluctance machine drives with smooth transition between control operations. *IEEE Trans. Autom. Control* **24**(1), 301–306 (2009)
6. Kulkarni, R.V., Venayagamoorthy, G.K.: Bio-inspired algorithms for autonomous deployment and localization of sensor nodes. *IEEE Trans. Syst. Man Cybern. Part C Appl. Rev.* **40**(6), 663–675 (2010)

7. Landelius, T.: Reinforcement learning and distributed local model synthesis. Ph.D. dissertation, Linköping University, Sweden (1997)
8. Lin, X.F., Zhang, H., Song, S.J., Song, C.N.: Adaptive dynamic programming with ϵ -error bound for nonlinear discrete-time systems. *J. Control Decis.* **26**(10), 1586–1590 (2011)
9. Lin, X.F., Huang, Y.J., Song, C.N.: Approximate optimal control with ϵ -error bound. *J. Control Theor. Appl.* **29**(1), 104–108 (2012)
10. Lin, X.F., Cao, N.Y., Lin, Y.U.: Optimal control for a class of nonlinear systems with state delay based on adaptive dynamic programming with ϵ -error bound. In: *Proceedings of IEEE Symposium Adaptive Dynamic Programming And Reinforcement Learning (ADPRL 2013)*, Singapore, pp 170–175 (2013)
11. Lin, X.F., Cao, N.Y., Song, S.J.: Optimal tracking control for a class of discrete-time nonlinear systems based on ϵ -ADP. *J. Guangxi Univ. Nat. Sci.* **39**(2), 372–377 (2014)
12. Lin, X.F., Ding, Q.: Adaptive dynamic programming optimal control based on approximation error of critic network. *J. Control Decis.* **30**(3), 495–499 (2015)
13. Liu, D., Zhang, Y., Zhang, H.: A self-learning call admission control scheme for CDMA cellular networks. *IEEE Trans. Neural Netw.* **16**(5), 1219–1228 (2005)
14. Mao, J., Cassandras, C.G.: Optimal control of multi-stage discrete event systems with real-time constraints. *IEEE Trans. Autom. Control* **54**(1), 108–123 (2009)
15. Murray, J.J., Cox, C.J., Lendaris, G.G., Saeks, R.: Adaptive dynamic programming. *IEEE Trans. Syst. Man Cybern. Part C Appl. Rev.* **32**(2), 140–153 (2002)
16. Necoara, I., Kerrigan, E.C., Schutter, B.D., Boom, T.: Finite-horizon min-max control of max-plus-linear systems. *IEEE Trans. Autom. Control* **52**(6), 1088–1093 (2007)
17. Prokhorov, D.V., Wunsch, D.C.: Adaptive critic designs. *IEEE Trans. Neural Netw.* **8**(5), 997–1007 (1997)
18. Si, J., Wang, Y.T.: On-line learning control by association and reinforcement. *IEEE Trans. Neural Netw.* **12**(2), 264–276 (2001)
19. Uchida, K., Fujita, M.: Finite horizon H^∞ control problems with terminal penalties. *IEEE Trans. Autom. Control* **37**(11), 1762–1767 (1992)
20. Vamvoudakis, K.G., Lewis, F.L.: Multi-player non-zero-sum games: Online adaptive learning solution of coupled Hamilton-Jacobi equations. *Automatica* **47**(8), 1556–1569 (2011)
21. Vrabie, D., Lewis, F.L.: Neural network approach to continuous-time direct adaptive optimal control for partially unknown nonlinear systems. *Neural Netw.* **22**(3), 237–246 (2009)
22. Wang, F.Y., 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.* **22**(1), 24–36 (2011)
23. Wang, F.Y., Zhang, H., Liu, D.: Adaptive dynamic programming: An introduction. *IEEE Comput. Intell. Mag.* **4**(2), 39–47 (2009)
24. Wei, Q., Liu, D.: Finite horizon optimal control of discrete-time nonlinear systems with unfixed initial state using adaptive dynamic programming. *J. Control Theor. Appl.* **9**(1), 123–133 (2011)
25. Wei, Q., Liu, D.: Optimal control for discrete-time nonlinear systems with unfixed initial state using adaptive dynamic programming. In: *Proceeding of International Joint Conference on Neural Networks (IJCNN 2011)*, pp. 61–67. USA, San Jose (2011)
26. Wei, Q., Liu, D.: An iterative ϵ -optimal control scheme for a class of discrete-time nonlinear systems with unfixed initial state. *Neural Netw.* **32**, 236–244 (2012)
27. Wei, Q., Zhang, H., Dai, J.: Model-free multiobjective approximate dynamic programming for discrete-time nonlinear systems with general performance index functions. *Neurocomputing* **72**(7–9), 1839–1848 (2009)
28. Werbos, P.J.: Advanced forecasting methods for global crisis warning and models of intelligence. *Gen. Syst. Yearb.* **22**, 25–38 (1977)
29. Werbos, P.J.: A menu of designs for reinforcement learning over time. In: Miller, W.T., Sutton, R.S., Werbos, P.J. (eds.) *Neural Netw. Control*. MIT Press, Cambridge (1991)
30. Werbos, P.J.: Intelligence in the brain: A theory of how it works and how to build it. *Neural Netw.* **22**(3), 200–212 (2009)

31. Zhang, H., Wei, Q., Liu, D.: An iterative adaptive dynamic programming method for solving a class of nonlinear zero-sum differential games. *Automatica* **47**(1), 207–214 (2011)
32. Zhang, H.G., Luo, Y.H., 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.* **20**(9), 1490–1503 (2009)
33. Zhang, H.G., Wei, Q., Luo, Y.H.: 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* **38**(4), 937–942 (2008)

Self-Learning Optimal Control of Nonlinear Systems

Adaptive Dynamic Programming Approach

Wei, Q.; Song, R.; Li, B.; Lin, X.

2018, XVIII, 230 p. 86 illus., 73 illus. in color., Hardcover

ISBN: 978-981-10-4079-5