A New Approach to Finite-Horizon Optimal Control for Discrete-Time Affine Nonlinear Systems via a Pseudolinear Method

Qinglai Wei, *Senior Member, IEEE*, Liao Zhu, Tao Li, and Derong Liu, *Fellow, IEEE*

***Abstract*—In this article, a new time-varying adaptive dynamic programming (ADP) algorithm is developed to solve finite-horizon optimal control problems for a class of discrete-time affine nonlinear systems. Inspired by the pseudolinear method, the nonlinear system can be approximated by a series of time-varying linear systems. In each iteration of the time-varying ADP algorithm, the optimal control law for the time-varying linear system is obtained. For an arbitrary initial state, it is proven that states of the time-varying linear systems converge to the states of discrete-time affine nonlinear systems. It is also shown that the iterative value functions and the iterative control laws converge to the optimal value function and the optimal control law, respectively. Finally, numerical results are presented to verify the effectiveness of the present method.**

***Index Terms*—Adaptive dynamic programming, approximate dynamic programming, finite horizon, nonlinear systems, optimal control, pseudolinear approximation.**

I. Introduction

Adaptive dynamic programming (ADP), proposed in [1], is an effective technique to solve optimal control problems and has gained much attention [2]–[5]. However, there exist some disadvantages for the neural network implementations of traditional ADP methods [6]–[8]. First, approximation via neural networks requires the collection of arrays of state and control data for training neural networks. Second, structures of neural networks are difficult to determine and the convergence of neural networks is difficult to analyze. Third, neural networks often result in huge computation time, especially for large-scale data approximation.

On the other hand, it is worth pointing out that the iterative control laws and the iterative value functions in each iteration of traditional ADP methods are generally nonanalytical functions, so that neural networks are used for numerical approximations. In [9], for continuous-time nonlinear systems $\dot{x} = A(x)x + B(x)u$, the "pseudolinear" method is used to develop a near-optimal control law. This provides a simple and efficient nonlinear design method by extending the principles of linear-quadratic regulator theory to continuous-time affine nonlinear systems [10]–[13]. In each iteration of the pseudolinear method, the analytical expressions of the time-varying iterative control laws and the iterative value functions can be obtained. It implies that the optimal control law and optimal value function can be approximated by a series of analytical time-varying iterative control laws and iterative value functions, if the Riccati equation is solvable in each iteration.

Many physical systems are described by continuous-time equations. They need to be discretized in order to develop an effective controller based on modern computer control technology implemented using microprocessors. One of the goals of this article is to develop an optimal control approach based on the discrete-time pseudolinear method, which can eliminate the use of neural networks in traditional ADP methods to obtain analytical expressions of the iterative control laws and the iterative value functions.

In this article, inspired by the continuous-time pseudolinear methods [9]–[13], a time-varying adaptive dynamic programming algorithm is developed, which can solve the finite-horizon optimal control problem for a class of discrete-time affine nonlinear systems. Main contributions of this article include the following.

1) It is proven that discrete-time affine nonlinear systems are approximated by a series of time-varying linear systems.
2) The analytical expressions of the iterative control laws and the iterative value functions are given. As the iteration index increases, the iterative control laws and the iterative value functions converge to the optimal control law and the optimal value function, respectively.
3) Detailed implementation of the present method is demonstrated with simulation results.

II. Problem Formulation

Consider a class of discrete-time affine nonlinear systems

$$\begin{aligned} x_{k+1} &= F(x_k, u_k) \\ &= (I + \Delta T A(x_k))x_k + \Delta T B(x_k)u_k \end{aligned} \tag{1}$$

where $x_k \in \mathbb{R}^n$ is the state and $u_k \in \mathbb{R}^m$ is the control input. Let $A(\cdot)$ and $B(\cdot)$ be the system functions. Let x_0 be the initial state. $\Delta T > 0$ is the sampling time interval. Let $\underline{u}_k^{\mathcal{T}_f-1} = \{u_k, u_{k+1}, \ldots, u_{\mathcal{T}_f-1}\}$ be an arbitrary control sequence from k to $\mathcal{T}_f - 1$, where the terminal time $\mathcal{T}_f$ is a positive integer. The value function is defined as

$$J_k(x_k, \underline{u}_k^{\mathcal{T}_f-1}) = \Psi(x_{\mathcal{T}_f}) + \sum_{\tau=k}^{\mathcal{T}_f-1} U(x_\tau, u_\tau) \tag{2}$$

where $U(x_\tau, u_\tau)$ is the utility function. It is expressed as $U(x_\tau, u_\tau) = x_\tau^{\mathrm{T}} Q(x_\tau)x_\tau + u_\tau^{\mathrm{T}} R(x_\tau)u_\tau$ with $Q(x_\tau) \geq 0$ and $R(x_\tau) > 0$ for $\forall x_\tau$,

Manuscript received March 29, 2021; accepted May 30, 2021. Date of publication June 8, 2021; date of current version April 26, 2022. This work was supported in part by the National Key Research and Development Program of China under Grant 2018YFB1702300, in part by the National Natural Science Foundation of China under Grants 62073321, 62073085, and in part by the Guangdong Introducing Innovative and Enterpreneurial Teams of "The Pearl River Talent Recruitment Program" under Grant 2019ZT08X340. Recommended by Associate Editor S. S. Saab. *(Corresponding author: Qinglai Wei.)*

Qinglai Wei, Liao Zhu, and Tao Li are with the State Key Laboratory for Management and Control of Complex Systems, Institute of Automation, Chinese Academy of Sciences, Beijing 100190, China, with the School of Artificial Intelligence, University of Chinese Academy of Sciences, Beijing 100049, China, also with the Institute of Systems Engineering, Macau University of Science and Technology, Macau 999078, China (e-mail: qinglai.wei@ia.ac.cn; liao.zhu@ia.ac.cn; litao2019@ia.ac.cn).

Derong Liu is with the School of Automation, Guangdong University of Technology, Guangzhou 510006, China (e-mail: derong@gdut.edu.cn).

Color versions of one or more figures in this article are available at https://doi.org/10.1109/TAC.2021.3087452.

Digital Object Identifier 10.1109/TAC.2021.3087452

0018-9286 © 2021 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission.
See https://www.ieee.org/publications/rights/index.html for more information.

Authorized licensed use limited to: ShanghaiTech University. Downloaded on June 04,2024 at 07:12:00 UTC from IEEE Xplore. Restrictions apply.

respectively. Let $\Psi(x_{\mathcal{T}_f}) = x_{\mathcal{T}_f}^{\mathsf{T}}\Gamma(x_{\mathcal{T}_f})x_{\mathcal{T}_f}$ be the terminal cost function, where $\Gamma(x_{\mathcal{T}_f}) \geq 0$ for $x_{\mathcal{T}_f}$. Results of this article are based on the following assumptions.

Assumption 1: The system (1) is controllable. For any $x_k, y_k \in \mathbb{R}^n$, assume that $A(\cdot)$, $B(\cdot)$, $Q(\cdot)$, $R(\cdot)$, and $\Gamma(\cdot)$ are Lipschitz continuous functions, such that

1) $\|A(x_k)-A(y_k)\| \leq \mathcal{L}_A\|x_k-y_k\|$
2) $\|B(x_k)-B(y_k)\| \leq \mathcal{L}_B\|x_k-y_k\|$
3) $\|Q(x_k)-Q(y_k)\| \leq \mathcal{L}_Q\|x_k-y_k\|$
4) $\|R(x_k)-R(y_k)\| \leq \mathcal{L}_R\|x_k-y_k\|$
5) $\|\Gamma(x_k)-\Gamma(y_k)\| \leq \mathcal{L}_\Gamma\|x_k-y_k\|$

where $\|\cdot\|$ denotes the Euclidean norm, and the parameters $\mathcal{L}_{(\cdot)}$ are positive constants.

For any state x_k, we attempt to find an optimal control law $\mu_k(x_k)$, such that the value function is minimized. For a fixed control law μ of (1), the map from state x_k to (2) is called a value function $J_k^\mu(x_k)$. The optimal value function is defined as $J_k^*(x_k) = \inf_\mu J_k^\mu(x_k), \forall k = 0, 1, \ldots, \mathcal{T}_f$. According to Bellman's principle of optimality, for $k = 0, 1, \ldots, \mathcal{T}_f - 1$, $J_k^*(x_k)$ satisfies the following discrete-time Bellman equation:

$$J_k^*(x_k) = \inf_{u_k}\left\{U(x_k, u_k) + J_{k+1}^*(x_{k+1})\right\}. \tag{3}$$

The terminal cost is defined as $J_{\mathcal{T}_f}^*(x_{\mathcal{T}_f}) = x_{\mathcal{T}_f}^{\mathsf{T}}\Gamma(x_{\mathcal{T}_f})x_{\mathcal{T}_f}$. For $k = 0, 1, \ldots, \mathcal{T}_f - 1$, we define the optimal control law as

$$u_k^*(x_k) = \arg\inf_{u_k}\left\{U(x_k, u_k) + J_{k+1}^*(x_{k+1})\right\}. \tag{4}$$

Hence, the Bellman equation (3) for $k = 0, 1, \ldots, \mathcal{T}_f - 1$ can be rewritten as

$$J_k^*(x_k) = U(x_k, u_k^*(x_k)) + J_{k+1}^*(F(x_k, u_k^*(x_k))). \tag{5}$$

Remark 1: Many physical systems are described by continuous-time dynamics but are controlled by discrete-time controllers. In order to get the discrete-time affine nonlinear systems (1), continuous-time affine nonlinear systems in [10] and [11] with the form $\dot{\chi} = A(\chi)\chi + B(\chi)\nu$ are discretized in this article, where χ is the state and ν is the control input. Considering a sampling time interval $\Delta T > 0$, we have $\dot{\chi} = \frac{\chi(t+\Delta T)-\chi(t)}{\Delta T}$. Then, we can derive $\chi_{(k+1)\Delta T} = (I + \Delta T A(\chi_{k\Delta T}))\chi_{k\Delta T} + \Delta T B(\chi_{k\Delta T})\nu_{k\Delta T}$. Let x_k and u_k, where $k = 0, 1, \ldots$, be the system state and control input, such that $x_k = \chi_{k\Delta T}$ and $u_k = \nu_{k\Delta T}$, respectively. Then, the discrete-time affine nonlinear systems can be expressed as (1).

III. Time-Varying Adaptive Dynamic Programming Algorithm

In this section, a new time-varying ADP method will be introduced to obtain the finite-horizon optimal control law for discrete-time affine nonlinear systems (1).

A. Derivation of the Time-Varying Value Function

According to the form of discrete-time affine nonlinear systems (1), for the iteration index $i = 0$, the initial linear system can be expressed as

$$x_{k+1}^{[0]} = (I + \Delta T A(x_0))x_k^{[0]} + \Delta T B(x_0)u_k^{[0]} \tag{6}$$

where $x_0^{[0]} = x_0$. The iterative value function is expressed as

$$V_k^{[0]}(x_k^{[0]}) = x_{\mathcal{T}_f}^{[0]\mathsf{T}}\Gamma(x_0)x_{\mathcal{T}_f}^{[0]} + \sum_{\tau=k}^{\mathcal{T}_f-1}(x_\tau^{[0]\mathsf{T}}Q(x_0)x_\tau^{[0]} + u_\tau^{[0]\mathsf{T}}R(x_0)u_\tau^{[0]}). \tag{7}$$

For $i = 1, 2, \ldots$, the time-varying linear system equation is expressed as

$$x_{k+1}^{[i]} = (I + \Delta T A(x_k^{[i-1]}))x_k^{[i]} + \Delta T B(x_k^{[i-1]})u_k^{[i]} \tag{8}$$

where $x_0^{[i]} = x_0$. The corresponding iterative value function is expressed as

$$V_k^{[i]}(x_k^{[i]}) = x_{\mathcal{T}_f}^{[i]\mathsf{T}}\Gamma(x_{\mathcal{T}_f}^{[i-1]})x_{\mathcal{T}_f}^{[i]} + \sum_{\tau=k}^{\mathcal{T}_f-1}(x_\tau^{[i]\mathsf{T}}Q(x_\tau^{[i-1]})x_\tau^{[i]} + u_\tau^{[i]\mathsf{T}}R(x_\tau^{[i-1]})u_\tau^{[i]}). \tag{9}$$

B. Iterative Control Law and the Closed-Loop System

According to (6)–(9), for any $i = 0, 1, \ldots$, the iterative systems are discrete-time time-varying linear systems and the value functions are expressed in time-varying quadratic forms. In this situation, the iterative value function in (7) and (9) for $i = 0, 1, \ldots$, can be described by a quadratic form [14], which is expressed as

$$V_k^{[i]}(x_k^{[i]}) = x_k^{[i]\mathsf{T}}P_k^{[i]}x_k^{[i]} \quad \forall k = 0, 1, \ldots, \mathcal{T}_f \tag{10}$$

and when $k = \mathcal{T}_f$, $P_{\mathcal{T}_f}^{[0]} = \Gamma(x_0)$ for $i = 0$ and $P_{\mathcal{T}_f}^{[i]} = \Gamma(x_{\mathcal{T}_f}^{[i-1]})$ for $i = 1, 2, \ldots$.

For $i = 0, 1, 2, \ldots$, let $v_k^{[i]}(x_k^{[i]})$ denote the pseudolinear feedback iterative control law (iterative control law in brief). Then, according to the principle of optimality [14], [15], the iterative control law $v_k^{[i]}(x_k^{[i]})$ $\forall k = 0, 1, \ldots, \mathcal{T}_f - 1$ is expressed as

$$\begin{aligned} v_k^{[i]}(x_k^{[i]}) &= -\Delta T\Big(R(x_k^{[i-1]}) + \Delta T^2 B^{\mathsf{T}}(x_k^{[i-1]})P_{k+1}^{[i]}B(x_k^{[i-1]})\Big)^{-1} \\ &\quad \times B^{\mathsf{T}}(x_k^{[i-1]})P_{k+1}^{[i]}(I + \Delta T A(x_k^{[i-1]}))x_k^{[i]} \end{aligned} \tag{11}$$

where $P_k^{[i]}$, $k = 0, 1, \ldots, \mathcal{T}_f - 1$ and $i = 0, 1, 2, \ldots$, in the iterative value function (10), which satisfies the following Riccati equation:

$$\begin{aligned} P_k^{[i]} &= Q(x_k^{[i-1]}) + (I + \Delta T A(x_k^{[i-1]}))^{\mathsf{T}}P_{k+1}^{[i]} \\ &\quad \times (I + \Delta T A(x_k^{[i-1]})) - \Delta T^2(I + \Delta T A(x_k^{[i-1]}))^{\mathsf{T}} \\ &\quad \times P_{k+1}^{[i]}B(x_k^{[i-1]})\Big(R(x_k^{[i-1]}) + \Delta T^2 B^{\mathsf{T}}(x_k^{[i-1]})P_{k+1}^{[i]} \\ &\quad \times B(x_k^{[i-1]})\Big)^{-1}B^{\mathsf{T}}(x_k^{[i-1]})P_{k+1}^{[i]}(I + \Delta T A(x_k^{[i-1]})) \end{aligned} \tag{12}$$

with the terminal constraints $P_{\mathcal{T}_f}^{[0]} = \Gamma(x_0)$ and $P_{\mathcal{T}_f}^{[i]} = \Gamma(x_{\mathcal{T}_f}^{[i-1]})$. For $i = 1, 2, \ldots$, the feedback system is expressed as

$$\begin{aligned} x_{k+1}^{[i]} &= (I + \Delta T A(x_k^{[i-1]}))x_k^{[i]} - \Delta T^2 B(x_k^{[i-1]})\Big(R(x_k^{[i-1]}) \\ &\quad + \Delta T^2 B^{\mathsf{T}}(x_k^{[i-1]})P_{k+1}^{[i]}B(x_k^{[i-1]})\Big)^{-1}B^{\mathsf{T}}(x_k^{[i-1]}) \\ &\quad \times P_{k+1}^{[i]}(I + \Delta T A(x_k^{[i-1]}))x_k^{[i]} \end{aligned} \tag{13}$$

where $x_k^{[i-1]} = x_0$ for $i = 0$, $k = 0, 1, \ldots, \mathcal{T}_f - 1$. The time-varying adaptive dynamic programming algorithm is summarized in Algorithm 1.

Authorized licensed use limited to: ShanghaiTech University. Downloaded on June 04,2024 at 07:12:00 UTC from IEEE Xplore. Restrictions apply.

Algorithm 1: The Time-Varying ADP Algorithm.

Initialization:

Give an initial state x_0.
Set a sampling time interval ΔT.
Give the terminal time $\mathcal{T}_f$.
Define a value function $J_k(x_k, \underline{u}_k^{\mathcal{T}_f-1})$ in (2).
Choose a positive integer $i_{\max}$.
Choose a positive number $\bar{T} > 1$.
Choose a computation precision ε.

Iteration:

1: Let the iteration index $i = 0$;
2: Obtain the control system (6).
3: Obtain $P_k^{[0]}$, $k = 0, 1, \ldots, \mathcal{T}_f - 1$ by solving the Riccati equation (12) with the terminal constraint $P_{\mathcal{T}_f}^{[0]} = \Gamma(x_0)$.
4: Solve the iterative control law $v_k^{[0]}(x_k^{[0]})$ with (11) and obtain the feedback system (13).
5: Record the trajectory of the state $x_k^{[0]}$ and $V_k^{[0]}(x_k^{[0]})$, where $k = 0, 1, \ldots, \mathcal{T}_f$.
6: Let $i = i + 1$ and go to the next step.
7: Obtain the discrete-time time-varying linear system (8) with the initial state x_0.
8: Obtain $P_k^{[i]}$, $k = 0, 1, \ldots, \mathcal{T}_f - 1$ by solving the Riccati equation (12) with the terminal constraint $P_{\mathcal{T}_f}^{[i-1]} = \Gamma(x_{\mathcal{T}_f}^{[i-1]})$.
9: Solve the iterative control law $v_k^{[i]}(x_k^{[i]})$ with (11) and obtain the feedback system (13).
10: If $\|x_k^{[i]} - x_k^{[i-1]}\| \le \varepsilon$, $k = 0, 1, \ldots, \mathcal{T}_f$, then go to Step 13. Else, go to Step 11.
11: If $i < i_{\max}$, then go to Step 12. Else, let $\Delta T = \dfrac{\Delta T}{\bar{T}}$ and go to Step 12.
12: Record the trajectory of the state $x_k^{[i]}$ and $V_k^{[i]}(x_k^{[i]})$, $\forall k = 0, 1, \ldots, \mathcal{T}_f$ and go to Step 6.
13: **return** ΔT, $v_k^{[i]}(x_k^{[i]})$, and $V_k^{[i]}(x_k^{[i]})$.

C. Properties of the Time-Varying Adaptive Dynamic Programming Algorithm

In this section, properties of the time-varying ADP algorithm will be discussed. Before the analysis, a notation will be defined. For $i = 0, 1, 2, \ldots$ and for k_0, such that $0 \le k_0 < k + 1$, let $\Phi^{[i-1]}(k+1, k_0)$ denote the transition matrix generated by $I + \Delta T A(x_k^{[i-1]})$ in (6) and (8), which can be expressed as

$$\Phi^{[i-1]}(k+1, k_0) = \prod_{\tau=k_0}^{k} \left(I + \Delta T A(x_\tau^{[i-1]})\right) \tag{14}$$

where $x_\tau^{[i-1]} = x_0$ for $i = 0$. Then, according to the definition of the transition matrices in (14), we can derive the following lemmas.

Lemma 1: For $i = 0, 1, \ldots$, and for an initial time $0 \le k_0 < k + 1$, let $\Phi^{[i-1]}(k+1, k_0)$ be the transition matrices defined in (14). Then, for any given $x_0 \in \mathbb{R}^n$, there exists a positive constant $\mathcal{A} > 0$, such that

$$\|\Phi^{[i-1]}(k+1, k_0)\| \le (1 + \Delta T \mathcal{A})^{k-k_0+1}. \tag{15}$$

Lemma 2: For $i = 0, 1, \ldots$, and $k = 0, 1, \ldots, \mathcal{T}_f$, let the time-varying ADP algorithm be implemented by (6)–(11) with an initial state x_0. Consider the Riccati equation (12) with the definite state x_0, there is an upper bound for the norm of the matrix $P_k^{[i]}$, where $i = 0, 1, \ldots$, and $k = 0, 1, \ldots, \mathcal{T}_f$, i.e., $\|P_k^{[i]}\| \le \mathcal{L}_P$, where $\mathcal{L}_P$ is a positive constant.

Proof: First, for $i = 0$, according to $P_{\mathcal{T}_f}^{[0]} = \Gamma(x_0)$, it can be derived that the norm of $P_{\mathcal{T}_f-1}^{[0]}$ in the Riccati equation (12) is bounded, as $A(x_k)$, $B(x_k)$, $Q(x_k)$, $R(x_k)$, and $\Gamma(x_k)$ are Lipschitz continuous. According to mathematical induction, we have that the norm of $P_k^{[0]}$, $k = 0, 1, \ldots, \mathcal{T}_f - 2$, is bounded. Then, it can be derived that the norm of $P_k^{[0]}$, $\forall k = 0, 1, \ldots, \mathcal{T}_f$, is bounded, i.e., $\|P_k^{[0]}\| \le \mathcal{L}_P^{[0]}$, where $\mathcal{L}_P^{[0]}$ is a positive constant.

Second, for $i = 1$, we have that the norm of $P_{\mathcal{T}_f}^{[1]} = \Gamma(x_{\mathcal{T}_f}^{[0]})$ is bounded. Using mathematical induction, it can be proven that the norm of $P_k^{[1]}$, $\forall k = 0, 1, \ldots, \mathcal{T}_f$, is bounded, i.e., $\|P_k^{[1]}\| \le \mathcal{L}_P^{[1]}$, where $\mathcal{L}_P^{[1]}$ is a positive constant.

Third, according to mathematical induction, it is proven that the norm of $P_k^{[i]}$, $i = 0, 1, \ldots$, and $k = 0, 1, \ldots, \mathcal{T}_f$, in Riccati equation (12) is bounded with an initial state x_0, i.e., $\|P_k^{[i]}\| \le \mathcal{L}_P$, where $\mathcal{L}_P = \max\{\mathcal{L}_P^{[i]}\}$. The proof is completed. ■

Remark 2: Given a deterministic initial state x_0, the iterative state $x_{k+1}^{[i]}$ of the time-varying linear system (8) is finite in a finite time step $\mathcal{T}_f$. According to Assumption 1, $P_{\mathcal{T}_f}^{[i]} = \Gamma(x_{\mathcal{T}_f}^{[i-1]})$, $i = 1, 2, \ldots$, is finite. In addition, according to (9), we have $V_{k+1}^{[i]}(x_{k+1}^{[i]}) - V_k^{[i]}(x_k^{[i]}) = -U(x_k^{[i]}, u_k^{[i]})$. The iterative value function $V_k^{[i]}(x_k^{[i]})$ (10) is decreasing for $(x_k^{[i]}, u_k^{[i]}) \neq 0$, as k increases. Thus, the norm of the matrix $P_k^{[i]}$, $i = 0, 1, \ldots$ and $k = 0, 1, \ldots, T_f$ is bounded.

We are now in a position to prove the following theorem.

Theorem 1: For $i = 0, 1, \ldots$, let $\mathcal{A} > 0$ be a positive constant, such that the transition matrix $\Phi^{[i-1]}(k+1, k_0)$ satisfies (15). Then, for $i = 1, 2, \ldots$, we have

$$\begin{aligned}\|\Phi^{[i-1]}(k+1, k_0) - \Phi^{[i-2]}(k+1, k_0)\| \\ \le\ & \mathcal{L}_A \Delta T (k - k_0 + 1)(1 + \Delta T \mathcal{A})^{k-k_0} \\ & \times \|x_0\| \sup_{s \in [k, k_0]} \|x_s^{[i-1]} - x_s^{[i-2]}\|. \end{aligned} \tag{16}$$

Proof: For $i = 0, 1, \ldots$, consider the following zero-input systems:

$$x_{k+1}^{[i]} = (I + \Delta T A(x_k^{[i-1]}))x_k^{[i]}, x_{k_0}^{[i]} = x_0 \tag{17}$$

and

$$x_{k+1}^{[i-1]} = (I + \Delta T A(x_k^{[i-2]}))x_k^{[i-1]}, x_{k_0}^{[i-1]} = x_0. \tag{18}$$

It is obvious that $\Phi^{[i-1]}(k+1, k_0)$ and $\Phi^{[i-2]}(k+1, k_0)$ are the solutions of the systems (17) and (18), respectively. According to (17) and (18), we can derive

$$\begin{aligned} x_{k+1}^{[i]} - & x_{k+1}^{[i-1]} \\ &= (I + \Delta T A(x_k^{[i-1]}))x_k^{[i]} - (I + \Delta T A(x_k^{[i-2]}))x_k^{[i-1]} \\ &= (I + \Delta T A(x_k^{[i-1]}))(x_k^{[i]} - x_k^{[i-1]}) \\ &\quad + ((I + \Delta T A(x_k^{[i-1]})) - (I + \Delta T A(x_k^{[i-2]})))x_k^{[i-1]}. \end{aligned} \tag{19}$$

Considering the definition of the transition matrix $\Phi^{[i-1]}(k+1, k_0)$, $i = 0, 1, \ldots$, in (14), we can obtain

$$\begin{aligned} x_{k+1}^{[i]} - x_{k+1}^{[i-1]} \\ = \Phi^{[i-1]}(k+1, k_0)(x_{k_0}^{[i]} - x_{k_0}^{[i-1]}) \end{aligned}$$

Authorized licensed use limited to: ShanghaiTech University. Downloaded on June 04,2024 at 07:12:00 UTC from IEEE Xplore. Restrictions apply.

$$+\sum_{\tau=k_0}^{k}\Phi^{[i-1]}(k+1,\tau+1)\left((I+\Delta TA(x_\tau^{[i-1]}))\right.$$
$$\left.-\left(I+\Delta TA(x_\tau^{[i-2]})\right)\right)x_\tau^{[i-1]}$$
$$=\sum_{\tau=k_0}^{k}\Phi^{[i-1]}(k+1,\tau+1)\left((I+\Delta TA(x_\tau^{[i-1]}))\right.$$
$$\left.-(I+\Delta TA(x_\tau^{[i-2]}))\right)\Phi^{[i-2]}(\tau,k_0)x_{k_0}^{[i-1]}. \quad (20)$$

According to Assumption 1 and Lemma 1, we can obtain

$$\begin{aligned}&\|x_{k+1}^{[i]}-x_{k+1}^{[i-1]}\|\\ &\le\sum_{\tau=k_0}^{k}(I+\Delta T\mathcal{A})^{k-\tau}\left(\mathcal{L}_A\Delta T\sup_{s\in[k,k_0]}\|x_\tau^{[i-1]}-x_\tau^{[i-2]}\|\right)\\ &\quad\times\|x_0\|(I+\Delta T\mathcal{A})^{\tau-k_0}\\ &\le\mathcal{L}_A\Delta T(k-k_0+1)(1+\Delta T\mathcal{A})^{k-k_0}\\ &\quad\times\|x_0\|\sup_{s\in[k,k_0]}\|x_s^{[i-1]}-x_s^{[i-2]}\|. \quad (21)\end{aligned}$$

According to (21), the inequality (16) can be verified. ■

According to (13), for $i=1,2,\ldots$, defining

$$\begin{aligned}C(x_k^{[i-1]})=&-B(x_k^{[i-1]})\left(R(x_k^{[i-1]})+\Delta T^2B^{\mathsf{T}}(x_k^{[i-1]})P_{k+1}^{[i]}\right.\\ &\left.\times B(x_k^{[i-1]})\right)^{-1}B^{\mathsf{T}}(x_k^{[i-1]})P_{k+1}^{[i]}\\ &\times(I+\Delta TA(x_k^{[i-1]})) \quad (22)\end{aligned}$$

the feedback system in (13) can be expressed as

$$\begin{aligned}x_{k+1}^{[i]}&=(I+\Delta TA(x_k^{[i-1]}))x_k^{[i]}+\Delta T^2C(x_k^{[i-1]})x_k^{[i]}\\ x_0^{[i]}&=x_0. \quad (23)\end{aligned}$$

Then, the convergence property can be analyzed.

Theorem 2: For $i=0,1,\ldots$, and $k=0,1,\ldots,\mathcal{T}_f$, the time-varying ADP algorithm is implemented by (6)–(11) with a definite initial state x_0. Then, there exists a sampling time interval ΔT, such that the iterative system state $x_k^{[i]}$ converges to the state x_k in (1) under the feedback control law $v_k^{[i]}(x_k^{[i]})$, as $i\to\infty$, i.e.,

$$\lim_{i\to\infty}x_k^{[i]}=x_k, k=0,1,\ldots,\mathcal{T}_f. \quad (24)$$

Proof: This theorem is proven in three steps.

Step 1: According to (22), let

$$\begin{aligned}C(x_k)=&-B(x_k)(R(x_k)+\Delta T^2B^{\mathsf{T}}(x_k)P_{k+1}\\ &\times B(x_k))^{-1}B^{\mathsf{T}}(x_k)P_{k+1}(I+\Delta TA(x_k)). \quad (25)\end{aligned}$$

Based on Assumption 1 and Lemma 2, it is proven that $C(x_k)$ is a Lipschitz continuous function.

First, it is easy to get that $\Delta T^2B^{\mathsf{T}}(x_k)$ is a Lipschitz continuous function because

$$\begin{aligned}\|\Delta T^2B^{\mathsf{T}}(x_k)-\Delta T^2B^{\mathsf{T}}(y_k)\|&\le\Delta T^2\|(B(x_k)-B(y_k))^{\mathsf{T}}\|\\ &\le\mathcal{L}_{TB}\|x_k-y_k\| \quad (26)\end{aligned}$$

where $\mathcal{L}_{TB}=\Delta T^2\mathcal{L}_B$ is a positive constant.

Second, it is shown that $P_{k+1}B(x_k)$ is a Lipschitz continuous function because

$$\begin{aligned}\|P_{k+1}B(x_k)-P_{k+1}B(y_k)\|&\le\|P_{k+1}\|\,\|B(x_k)-B(y_k)\|\\ &\le\mathcal{L}_{PB}\|x_k-y_k\| \quad (27)\end{aligned}$$

where $\mathcal{L}_{PB}=\mathcal{L}_P\mathcal{L}_B$ is a positive constant.

Third, based on the proof in (26)–(27), we can prove that $\Delta T^2B^{\mathsf{T}}(x_k)P_{k+1}B(x_k)$, $R(x_k)+\Delta T^2B^{\mathsf{T}}(x_k)P_{k+1}B(x_k)$ and $P_{k+1}(I+\Delta TA(x_k))$ are all Lipschitz continuous functions for $x_k,y_k\in\mathbb{R}^n$.

Based on the above analysis, we can derive that $C(x_k)$ is a Lipschitz continuous function, such that

$$\|C(x_k)-C(y_k)\|\le\mathcal{L}_C\|x_k-y_k\|\quad\forall x_k,y_k\in\mathbb{R}^n. \quad (28)$$

Step 2: According to (22), based on Assumption 1 and Lemma 2, it is proven that the norm of $C(x_k^{[i-1]})$, where $i=0,1,\ldots$, and $k=0,1,\ldots,\mathcal{T}_f$, is bounded with an initial state x_0.

First, for $i=0$, according to $x_k^{[i-1]}=x_0$, where $k=0,1,\ldots,\mathcal{T}_f-1$, the norm of $C(x_k^{[i-1]})$ can be derived as

$$\begin{aligned}\|C(x_0)\|\le&\|B(x_0)\|\left\|\left(R(x_0)+\Delta T^2B^{\mathsf{T}}(x_0)P_{k+1}^{[0]}\right.\right.\\ &\left.\left.\times B(x_0)\right)^{-1}\right\|\left\|B^{\mathsf{T}}(x_0)\right\|\left\|P_{k+1}^{[0]}(I+\Delta TA(x_0))\right\|. \quad (29)\end{aligned}$$

Letting $\sigma_j^{[0]}$, $j=1,2,\ldots,n$, be the singular values of the matrix $R(x_0)+\Delta T^2B^{\mathsf{T}}(x_0)P_{k+1}^{[0]}B(x_0)$, then we have [16]

$$\left\|\left(R(x_0)+\Delta T^2B^{\mathsf{T}}(x_0)P_{k+1}^{[0]}B(x_0)\right)^{-1}\right\|=\frac{1}{\min_j\sigma_j^{[0]}}. \quad (30)$$

Letting $\bar\sigma_j^{[0]}=1/\min_j\sigma_j^{[0]}$, then $\|C(x_0)\|$ can be derived as

$$\begin{aligned}\|C(x_0)\|\le&\bar\sigma_j^{[0]}(\mathcal{L}_B\|x_0\|+\|B(0)\|)^2\mathcal{L}_P\\ &\times(1+\Delta T(\mathcal{L}_A\|x_0\|+\|A(0)\|)). \quad (31)\end{aligned}$$

It can easily be derived that $\|C(x_0)\|\le\mathcal{C}^{[0]}$, where $C^{[0]}=\bar\sigma_j^{[0]}(\mathcal{L}_B\|x_0\|+\|B(0)\|)^2\mathcal{L}_P(1+\Delta T(\mathcal{L}_A\|x_0\|+\|A(0)\|))$ is a positive constant.

Second, for $i=1$, according to $x_0^{[i]}=x_0$ and (29)–(31), the norm of $C(x_k^{[i-1]})$ can be derived as

$$\begin{aligned}\left\|C\left(x_k^{[0]}\right)\right\|\le&\bar\sigma_j^{[1]}\left(\mathcal{L}_B\left\|x_k^{[0]}\right\|+\|B(0)\|\right)^2\mathcal{L}_P\\ &\times\left(1+\Delta T\left(\mathcal{L}_A\left\|x_k^{[0]}\right\|+\|A(0)\|\right)\right)\end{aligned}$$

where $\bar\sigma_j^{[1]}=1/\min_j\sigma_j^{[1]}$, $j=1,2,\ldots,n$, and $\sigma_j^{[1]}$ are the singular values of matrix $R(x_k^{[0]})+\Delta T^2B^{\mathsf{T}}(x_k^{[0]})P_{k+1}^{[0]}B(x_k^{[0]})$. Then, it can easily be derived that $\|C(x_k^{[0]})\|\le\mathcal{C}^{[1]}$, where $C^{[1]}=\bar\sigma_j^{[1]}(1+\Delta T(\mathcal{L}_A\|x_k^{[0]}\|+\|A(0)\|))\mathcal{L}_P(\mathcal{L}_B\|x_k^{[0]}\|+\|B(0)\|)^2$ is a positive constant.

Third, according to mathematical induction, the norm of $C(x_k^{[i-1]})$, $i=0,1,\ldots$ and $k=0,1,\ldots,\mathcal{T}_f$, is upper bounded with the initial state x_0, such that

$$\|C(x_k^{[i-1]})\|\le\mathcal{C} \quad (32)$$

where $\mathcal{C}=\max\{\mathcal{C}^{[i]}\}$, $i=0,1,\ldots$, is a positive constant.

Step 3: Prove (24).

For $k=0,1,\ldots,\mathcal{T}_f-1$, considering the feedback control system (23), we have

Authorized licensed use limited to: ShanghaiTech University. Downloaded on June 04,2024 at 07:12:00 UTC from IEEE Xplore. Restrictions apply.

$$x_{k+1}^{[i]} - x_{k+1}^{[i-1]} = (I + \Delta T A(x_k^{[i-1]}))x_k^{[i]} + \Delta T^2 C(x_k^{[i-1]})x_k^{[i]} - (I + \Delta T A(x_k^{[i-2]}))x_k^{[i-1]} - \Delta T^2 C(x_k^{[i-2]})x_k^{[i-1]}. \quad (33)$$

According to the definition of $\Phi^{[i-1]}(k+1,k_0)$, $i=0,1,\ldots$, in (14), we can get

$$\begin{aligned} & x_{k+1}^{[i]} - x_{k+1}^{[i-1]} \\ &= (\Phi^{[i-1]}(k+1,0) - \Phi^{[i-2]}(k+1,0))x_0 \\ &\quad + \sum_{\tau=0}^{k} \Delta T^2 \Phi^{[i-1]}(k+1,\tau+1)C(x_k^{[i-1]})(x_\tau^{[i]} - x_\tau^{[i-1]}) \\ &\quad + \sum_{\tau=0}^{k} \Delta T^2 \Phi^{[i-1]}(k+1,\tau+1)(C(x_k^{[i-1]}) - C(x_k^{[i-2]}))x_\tau^{[i-1]} \\ &\quad + \sum_{\tau=0}^{k} \Delta T^2 (\Phi^{[i-1]}(k+1,\tau+1) - \Phi^{[i-2]}(k+1,\tau+1)) \\ &\quad \times C(x_k^{[i-1]})x_\tau^{[i-1]}. \end{aligned} \quad (34)$$

As $A(\cdot)$ is Lipschitz continuous for x_k, then for $i=0,1,\ldots$, the system state $x_{k+1}^{[i]}$ is finite under any initial state x_0, where we let $x_k^{[-1]} \equiv x_0$. Then, for $i=0,1,\ldots$ and $k_0 \geq 0$, there exists a positive number $\sigma > 0$, such that

$$\sup_{s\in[k,k_0]} \|x_s^{[i]} - x_s^{[i-1]}\| \leq \sigma \sup_{s\in[k,k_0]} \|x_{s+1}^{[i]} - x_{s+1}^{[i-1]}\|. \quad (35)$$

According to (23), we can obtain

$$\|x_{k+1}^{[i]}\| \leq (1 + \Delta T\mathcal{A} + \Delta T^2\mathcal{C})^{k+1}\|x_0\|. \quad (36)$$

According to (15), (16), (28), (32), and (34)–(36), it can be derived that

$$\begin{aligned} & \sup_{s\in[k,0]} \|x_{s+1}^{[i]} - x_{s+1}^{[i-1]}\| \\ &\leq \mathcal{L}_A \Delta T(k+1)(1+\Delta T\mathcal{A})^k \|x_0\| \sup_{s\in[k,0]} \|x_s^{[i-1]} - x_s^{[i-2]}\| \\ &\quad + \sum_{\tau=0}^{k} \left(\Delta T^2 \mathcal{C}(1+\Delta T\mathcal{A})^{k-\tau} \sup_{s\in[k,0]} \|x_s^{[i]} - x_s^{[i-1]}\| \right) \\ &\quad + \sum_{\tau=0}^{k} \Big(\Delta T^2 \mathcal{L}_C (1+\Delta T\mathcal{A})^{k-\tau} (1 + \Delta T\mathcal{A} + \Delta T^2\mathcal{C})^{\tau} \\ &\quad \times \|x_0\| \sup_{s\in[k,0]} \|x_s^{[i]} - x_s^{[i-1]}\| \Big) + \sum_{\tau=0}^{k} \Big(\mathcal{L}_A \mathcal{C} \Delta T^3 (k-\tau) \\ &\quad \times (1+\Delta T\mathcal{A})^{k-\tau-1} (1 + \Delta T\mathcal{A} + \Delta T^2\mathcal{C})^{\tau} \\ &\quad \times \|x_0\| \sup_{s\in[k,0]} \|x_k^{[i]} - x_k^{[i-1]}\| \Big). \end{aligned} \quad (37)$$

Letting $\xi_{k+1}^{[i]} = \sup_{s\in[k+1,0]} \|x_s^{[i]} - x_s^{[i-1]}\|$ and according to (35) and (37), we can obtain

$$\begin{aligned} & \left(1 - \Delta T^2 \mathcal{C}\sigma \sum_{\tau=0}^{k} (1+\Delta T\mathcal{A})^{k-\tau}\right) \xi_{k+1}^{[i]} \\ &\leq \mathcal{L}_A \Delta T \sigma (k+1)(1+\Delta T\mathcal{A})^k \|x_0\| \xi_{k+1}^{[i-1]} \\ &\quad + \Delta T^2 \mathcal{L}_C \sigma \|x_0\| \sum_{\tau=0}^{k} \Big((1+\Delta T\mathcal{A})^{k-\tau} \\ &\quad \times (1 + \Delta T\mathcal{A} + \Delta T^2\mathcal{C})^{\tau} \Big) \xi_{k+1}^{[i-1]} + \mathcal{L}_A \mathcal{C} \Delta T^3 \\ &\quad \times \sigma \|x_0\| \sum_{\tau=0}^{k} \Big((k-\tau)(1+\Delta T\mathcal{A})^{k-\tau-1} \\ &\quad \times (1 + \Delta T\mathcal{A} + \Delta T^2\mathcal{C})^{\tau} \Big) \xi_{k+1}^{[i-1]}. \end{aligned} \quad (38)$$

Then, we obtain

$$\xi_{k+1}^{[i]} \leq \eta_{k+1} \xi_{k+1}^{[i-1]} \quad (39)$$

where η_{k+1} is expressed as in (40) on the next page.

From (40), if we choose a small ΔT, such that $\eta_{k+1} < 1$, then according to (39), for $i \to \infty$, we have $\xi_{k+1}^{[i]} \to 0$, which implies that $x_{k+1}^{[i]}$ is convergent.

Letting $\lim_{i\to\infty} x_{k+1}^{[i]} = x_{k+1}^{[\infty]}$, the system (13) can be derived as

$$x_{k+1}^{[\infty]} = (I + \Delta T A(x_k^{[\infty]}))x_k^{[\infty]} + B(x_k^{[\infty]})v_k^{[\infty]}(x_k^{[\infty]}) \quad (41)$$

where $v_k^{[\infty]}(x_k^{[\infty]})$ is expressed as

$$\begin{aligned} & v_k^{[\infty]}(x_k^{[\infty]}) \\ &= -\Delta T^2 \left(R(x_k^{[\infty]}) + \Delta T^2 B^{\mathsf{T}}(x_k^{[\infty]}) P_{k+1}^{[\infty]} B(x_k^{[\infty]}) \right)^{-1} \\ &\quad \times B^{\mathsf{T}}(x_k^{[\infty]}) P_{k+1}^{[\infty]} (I + \Delta T A(x_k^{[\infty]}))x_k^{[\infty]}. \end{aligned} \quad (42)$$

Letting $x_{k+1} = x_{k+1}^{[\infty]}$ and $u_k = v_k^{[\infty]}$, we can obtain (1) for $k = 0,1,\ldots,\mathcal{T}_f - 1$, which implies $x_{k+1}^{[i]} \to x_{k+1}$ as $i \to \infty$. As $x_0^{[i]} = x_0$, $\forall i = 0,1,\ldots$, (24) can easily be derived, which implies (24) holds for all $k = 0,1,\ldots,\mathcal{T}_f$. The proof is completed. ■

Theorem 2 shows that the iterative system state $x_k^{[i]}$ of time-varying linear systems converges to the state x_k of the nonlinear system (1), as $i \to \infty$. In the following statement, the optimality of the time-varying ADP algorithm will be discussed.

Theorem 3: For $i = 0,1,\ldots$, let the time-varying ADP algorithm be implemented by (6)–(11). There exists a finite sampling time interval ΔT, such that the iterative value function $V_k^{[i]}(x_k^{[i]})$ converges to the optimal value function $J_k^*(x_k)$, as the iteration index i increases to infinity, i.e.,

$$\lim_{i\to\infty} V_k^{[i]}(x_k^{[i]}) = J_k^*(x_k), \ k = 0,1,\ldots,\mathcal{T}_f. \quad (43)$$

Proof: In the time-varying ADP algorithm (6)–(11), the system (1) is approximated by a series of time-varying linear systems which are expressed by (6) and (8), respectively, for $i = 0,1,\ldots$. Let $x_k^{[i-1]} = x_0$ for $i = 0$. Then, for any $i = 0,1,\ldots$, the iterative control laws $v_k^{[i]}(x_k^{[i]})$ in (11) are derived by the following equation:

$$\begin{aligned} v_k^{[i]}(x_k^{[i]}) = \ & \arg\min_{u_k^{[i]}} \Big\{ x_k^{[i]\mathsf{T}} Q(x_k^{[i-1]}) x_k^{[i]} + u_k^{[i]\mathsf{T}} R(x_k^{[i-1]}) u_k^{[i]} \\ & + V_{k+1}^{[i]}(x_{k+1}^{[i]}) \Big\} \end{aligned} \quad (44)$$

where

$$\begin{aligned} & V_k^{[i]}(x_k^{[i]}) = x_k^{[i]} P_k^{[i]} x_k^{[i]}, k = 0,1,\ldots,\mathcal{T}_f - 1 \\ & V_{\mathcal{T}_f}^{[i]}(x_{\mathcal{T}_f}^{[i]}) = x_{\mathcal{T}_f}^{[i]} \Gamma(x_{\mathcal{T}_f}^{[i-1]}) x_{\mathcal{T}_f}^{[i]} \end{aligned} \quad (45)$$

and $P_k^{[i]}$ satisfies the Riccati equation (12).

Authorized licensed use limited to: ShanghaiTech University. Downloaded on June 04,2024 at 07:12:00 UTC from IEEE Xplore. Restrictions apply.

For $i \to \infty$, choosing a small ΔT, according to Theorem 2, we have $x_k^{[i]} \to x_k^{[\infty]}$, as $i \to \infty$. For $i \to \infty$, define

$$\begin{aligned} V_k^{[\infty]}(x_k^{[\infty]}) &= x_k^{[\infty]} P_k^{[\infty]} x_k^{[\infty]}, k = 0, 1, \ldots, \mathcal{T}_f - 1 \\ V_{\mathcal{T}_f}^{[\infty]}(x_{\mathcal{T}_f}^{[\infty]}) &= x_{\mathcal{T}_f}^{[\infty]} \Gamma(x_{\mathcal{T}_f}^{[\infty]}) x_{\mathcal{T}_f}^{[\infty]} \end{aligned} \tag{46}$$

where $P_k^{[\infty]}$ satisfies the following Riccati equation:

$$\begin{aligned} P_k^{[\infty]} = {} & Q(x_k^{[\infty]}) + (I + \Delta T A(x_k^{[\infty]}))^{\mathsf{T}} P_{k+1}^{[\infty]} \\ & \times (I + \Delta T A(x_k^{[\infty]})) - \Delta T^2 (I + \Delta T A(x_k^{[\infty]}))^{\mathsf{T}} \\ & \times P_{k+1}^{[\infty]} B(x_k^{[\infty]}) \Big(R(x_k^{[\infty]}) + \Delta T^2 B^{\mathsf{T}}(x_k^{[\infty]}) P_{k+1}^{[\infty]} \\ & \times B(x_k^{[\infty]}) \Big)^{-1} B^{\mathsf{T}}(x_k^{[\infty]}) P_{k+1}^{[\infty]} (I + \Delta T A(x_k^{[\infty]})) \end{aligned} \tag{47}$$

with the terminal constraint $P_{\mathcal{T}_f}^{[\infty]} = \Gamma(x_{\mathcal{T}_f}^{[\infty]})$. According to Lemma 2, it is known that $V_k^{[\infty]}(x_k^{[\infty]})$ is finite for $k = 0, 1, \ldots, \mathcal{T}_f$, since the trajectory of $x_k^{[\infty]}, k = 0, 1, \ldots, \mathcal{T}_f$, is fixed. For $k = 0, 1, \ldots, \mathcal{T}_f - 1$, $v_k^{[\infty]}(x_k^{[\infty]})$ is defined as (42). According to (44) and (46), we can derive

$$\begin{aligned} v_k^{[\infty]}(x_k^{[\infty]}) = {} & \arg\min_{u_k^{[\infty]}} \Big\{ x_k^{[\infty]\mathsf{T}} Q(x_k^{[\infty]}) x_k^{[\infty]} \\ & + u_k^{[\infty]\mathsf{T}} R(x_k^{[\infty]}) u_k^{[\infty]} + V_{k+1}^{[\infty]}(x_{k+1}^{[\infty]}) \Big\}. \end{aligned} \tag{48}$$

Thus, $V_k^{[\infty]}(x_k^{[\infty]}), k = 0, 1, \ldots, \mathcal{T}_f - 1$, satisfies

$$\begin{aligned} & V_k^{[\infty]}(x_k^{[\infty]}) \\ & = x_k^{[\infty]\mathsf{T}} Q(x_k^{[\infty]}) x_k^{[\infty]} + v_k^{[\infty]\mathsf{T}}(x_k^{[\infty]}) R(x_k^{[\infty]}) v_k^{[\infty]}(x_k^{[\infty]}) \\ & \quad + V_{k+1}^{[\infty]} \Big((I + \Delta T A(x_k^{[\infty]}) + \Delta T^2 C(x_k^{[\infty]})) x_k^{[\infty]} \Big) \\ & = \min_{u_k^{[\infty]}} \Big\{ x_k^{[\infty]\mathsf{T}} Q(x_k^{[\infty]}) x_k^{[\infty]} + u_k^{[\infty]\mathsf{T}} R(x_k^{[\infty]}) u_k^{[\infty]} \\ & \quad + V_{k+1}^{[\infty]}(x_{k+1}^{[\infty]}) \Big\} \end{aligned} \tag{49}$$

which is the Bellman equation (3). For $k = \mathcal{T}_f$, the terminal constraint function satisfies

$$V_{\mathcal{T}_f}^{[\infty]}(x_{\mathcal{T}_f}^{[\infty]}) = x_{\mathcal{T}_f}^{[\infty]} \Gamma(x_{\mathcal{T}_f}^{[\infty]}) x_{\mathcal{T}_f}^{[\infty]} = J_{\mathcal{T}_f}^*(x_{\mathcal{T}_f}). \tag{50}$$

It shows that $J_k^*(x_k) = V_k^{[\infty]}(x_k^{[\infty]})$, for $k = 0, 1, \ldots, \mathcal{T}_f$. The proof is completed.

Corollary 1: For $i = 0, 1, \ldots$, let the time-varying ADP algorithm be implemented by (6)–(11). If the sampling time interval ΔT is small enough, then the iterative control law $v_k^{[i]}(x_k)$ converges to the optimal control law $u_k^*(x_k)$ in (3), as $i \to \infty$.

Remark 3: According to Theorem 3, for a given initial state x_0, the optimal value function $J_k^*(x_k), k = 0, 1, \ldots, \mathcal{T}_f$, can be approximated by the iterative value function $x_k^{[i]\mathsf{T}} P_k^{[i]} x_k^{[i]}$ as $i \to \infty$. However, it should be emphasized that it does not mean that $J_k^*(x_k)$ is a quadratic function for all $x_k \in \mathbb{R}^n$. For example, if the initial state x_0 is changed to $\bar{x}_0$, where $x_0 \neq \bar{x}_0$, then the iterative value function is also changed, such as $\bar{x}_k^{[i]\mathsf{T}} \bar{P}_k^{[i]} \bar{x}_k^{[i]}$. Generally speaking, $J_k^*(x_k)$, $\forall x_k \in \mathbb{R}^n$, is a nonanalytical function for nonlinear systems, which cannot be approximated by a single quadratic function for the entire state space. On the other hand, given an initial state x_0, it is declared that the Bellman equation is not solved for all $x_k \in \mathbb{R}^n$ by the time-varying ADP algorithm. Actually, for the initial state x_0, the time-varying ADP algorithm obtains a pointwise optimal solution of the affine nonlinear system (1), which is not the optimal solution for all $x_k \in \mathbb{R}^n$.

IV. Simulation Studies

In this section, simulation results are shown to verify the performance of our time-varying ADP algorithm. The optimal control problem for the inverted pendulum system [17] with modifications is considered, where the sinusoidal term is replaced by polynomial terms. $m = 1/2\,\mathrm{kg}$ and $\ell = 1/3\,\mathrm{m}$ are the mass and length of the pendulum bar, respectively. Let $\kappa = 0.2$ and $g = 9.8\,\mathrm{m/s^2}$ be the frictional factor and the gravitational acceleration, respectively. Let the approximation parameters be $\theta_0 = -6$ and $\theta_1 = 120$, respectively. Discretization of the system function with the sampling time interval ΔT leads to

$$\begin{aligned} \begin{bmatrix} x_{1(k+1)} \\ x_{2(k+1)} \end{bmatrix} = {} & \begin{bmatrix} 1 & \Delta T \\ \Delta T \frac{g}{\ell}(1 + \frac{1}{\theta_0} x_{1k}^2 + \frac{1}{\theta_1} x_{1k}^4) & 1 - \Delta T \kappa \ell \end{bmatrix} \\ & \times \begin{bmatrix} x_{1k} \\ x_{2k} \end{bmatrix} + \begin{bmatrix} 0 \\ \frac{\Delta T}{m\ell^2} \end{bmatrix} u_k. \end{aligned} \tag{51}$$

Let the value function be expressed by (2). Choose $Q = I_1$ and $R = I_2$, where I_1 and I_2 denote the identity matrices with suitable dimensions. Let $\mathcal{T}_f = 500$. Choose $\Gamma(x_k) = I_1$, $\forall x_k$. As Algorithm 1 cannot be implemented for infinite times to reach the convergence, a computation precision ε is given. If the inequality $\|x_k^{[i]} - x_k^{[i-1]}\| \leq \varepsilon$ is satisfied for $k = 0, 1, \ldots, \mathcal{T}_f$, then the state is regarded as convergent. It is required to use a small positive number ε in Algorithm 1, such that the state $x_k^{[i-1]}$ is sufficiently close to $x_k^{[i]}$. We choose $\varepsilon = 0.01$ and an initial state $x_0 = [5, 2]^{\mathsf{T}}$ to illustrate the effectiveness of our algorithm. Let $i_{\max} = 20$, the initial sampling time interval $\Delta T = 0.1$ and $\bar{T} = 10$. Implement the time-varying adaptive dynamic programming algorithm in Algorithm 1. The algorithm returns $\Delta T = 0.01$ and it takes 20 iterations to reach the computation precision.

The trajectories of the iterative value function $V_k^{[i]}(x_k^{[i]})$, which start with the initial state x_0, are shown in Fig. 1. The word "In" denotes initial iteration and the word "Lm" denotes the limiting iteration. It is shown that the iterative value functions $V_k^{[i]}(x_k^{[i]})$ converge to the optimum, as the iteration index i increases. Implementing the time-varying ADP algorithm with the initial state $x_0 = [5, 2]^{\mathsf{T}}$, the trajectories of states and controls are shown in Fig. 2(a)–(c), respectively, where iterative system states $x_k^{[i]}$ and control law $v_k^{[i]}(x_k^{[i]})$ both converge to their optimums.

In the implementation of the time-varying ADP algorithm, in order to obtain the iterative control law (11), we have to solve the Riccati equation (12) in each iteration. $P_k^{[i]} \in \mathbb{R}^{2\times 2}$ is the solution of the Riccati equation (12) associated with the nonlinear system (51), which

$$\begin{aligned} \eta_{k+1} = {} & \Delta T \|x_0\| \Bigg[\mathcal{L}_A \sigma (k+1)(1 + \Delta T \mathcal{A})^k x_0 + \Delta T \mathcal{L}_C \sigma \sum_{\tau=0}^{k} \Big((1 + \Delta T \mathcal{A})^{k-\tau} \big(1 + \Delta T \mathcal{A} + \Delta T^2 \mathcal{C}\big)^{\tau} \Big) \\ & + \mathcal{L}_A \mathcal{C} \Delta T^2 \sigma \sum_{\tau=0}^{k} \Bigg((k - \tau)(1 + \Delta T \mathcal{A})^{k-\tau-1} \big(1 + \Delta T \mathcal{A} + \Delta T^2 \mathcal{C}\big)^{\tau} \Bigg/ \Big(1 - \Delta T^2 \mathcal{C} \sigma \sum_{\tau=0}^{k} (1 + \Delta T \mathcal{A})^{k-\tau} \Big) \Bigg) \Bigg] \end{aligned} \tag{40}$$

Authorized licensed use limited to: ShanghaiTech University. Downloaded on June 04,2024 at 07:12:00 UTC from IEEE Xplore. Restrictions apply.

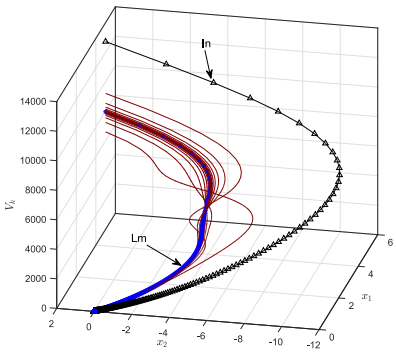

Fig. 1. Trajectories of the iterative value functions $V_k^{[i]}(x_k^{[i]})$ started by the initial state.

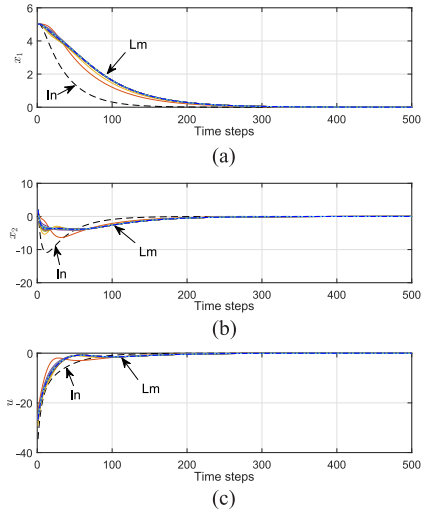

Fig. 2. State and control trajectories. (a) State trajectories of x_1. (b) State trajectories of x_2. (c) Control trajectories.

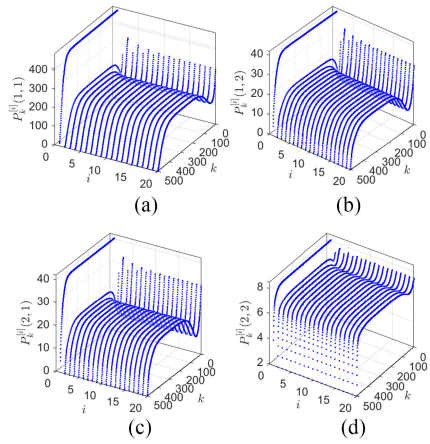

Fig. 3. Plots of the solution $P_k^{[i]}$ in the Riccati equation. (a) $P_k^{[i]}(1,1)$. (b) $P_k^{[i]}(1,2)$. (c) $P_k^{[i]}(2,1)$. (d) $P_k^{[i]}(2,2)$.

is defined as $P_k^{[i]} = [P_k^{[i]}(1,1), P_k^{[i]}(1,2); P_k^{[i]}(2,1), P_k^{[i]}(2,2)]$. The plots of $[P_k^{[i]}(1,1), P_k^{[i]}(1,2); P_k^{[i]}(2,1), P_k^{[i]}(2,2)]$ are shown in Fig. 3 . It is shown that the four elements of $P_k^{[i]}$ converge to the solutions of the Riccati equation, as the iteration index i increases. The optimal trajectories of the states and control are shown in Fig. 4(a)–(c), respectively.

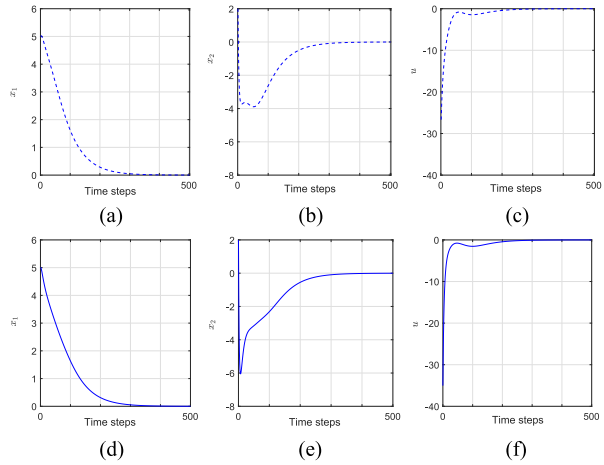

Fig. 4. State and control trajectories based on the time-varying ADP algorithm and the direct collocation methods. Dashed line: Time-varying ADP. Solid line: Direct collocation methods. (a) State trajectory of x_1. (b) State trajectory of x_2. (c) Control trajectory. (d) State trajectory of x_1. (e) State trajectory of x_2. (f) Control trajectory.

In order to show the effectiveness of the present ADP method, the numerical solution of the optimal control problem of the nonlinear system (51) is obtained by direct collocation methods [18], [19]. In detail, the continuous-time optimal control problem is formulated as a nonlinear programming problem [18]. Then, the resulting nonlinear programming problem is solved by the primal-dual Lagrange multiplier method [19]. In the comparison, the optimal control solver is implemented with the same initial conditions for the nonlinear system (51). The trajectories of the states and controls are shown in Fig. 4(d)–(f), respectively.

From Fig. 4, it is shown that the trajectories obtained by the time-varying ADP algorithm are similar to the ones by the direct collocation and primal-dual Lagrange multiplier methods, which verifies the effectiveness of the developed time-varying ADP algorithm. On the other hand, using the direct collocation, the optimal control problem of the nonlinear system is formulated as a nonlinear programming problem, which is solved by primal-dual Lagrange multiplier methods in an open loop. In contrast, the optimal control law by the time-varying ADP algorithm is a closed-loop control law. Furthermore, using the direct collocation and primal-dual Lagrange multiplier methods, the numerical solution of the nonlinear programming problem is obtained. It is declared that the control law by the time-varying ADP algorithm is an analytical one. Thus, superiorities of our ADP method can be verified.

V. Conclusion

In this article, a new discrete-time time-varying adaptive dynamic programming (ADP) is developed to solve the finite-horizon optimal control for a class of discrete-time affine nonlinear systems. Given an initial state, it is proven that the states of the time-varying pseudolinear systems converge to the ones of the nonlinear system. It is shown that the iterative value function and iterative control law converge to the optimal value function and the optimal control law, respectively, if the sampling time interval is small enough. The detailed implementation of the time-varying ADP algorithm has been provided.

In addition, it is pointed out that the present method can only be used to solve the optimal control problem of discretized affine nonlinear systems, but not the genuine discrete-time systems. In our future work, we will focus on the applications of the pseudolinear method to the genuine discrete-time systems and the stability of nonlinear systems.

Authorized licensed use limited to: ShanghaiTech University. Downloaded on June 04,2024 at 07:12:00 UTC from IEEE Xplore. Restrictions apply.

REFERENCES

[1] P. J. Werbos, "Advanced forecasting methods for global crisis warning and models of intelligence," *Gen. Syst. Yearbook*, vol. 22, pp. 25–38, Jan. 1977.

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

[3] D. Liu, S. Xue, B. Zhao, B. Luo, and Q. Wei, "Adaptive dynamic programming for control: A survey and recent advances," *IEEE Trans. Syst. Man Cybern. Syst.*, vol. 51, no. 1, pp. 142–160, Jan. 2021.

[4] Q. Wei, D. Liu, and H. Lin, "Value iteration adaptive dynamic programming for optimal control of discrete-time nonlinear systems," *IEEE Trans. Cybern.*, vol. 46, no. 3, pp. 840–853, Mar. 2016.

[5] W. Gao, Y. Jiang, Z.-P. Jiang, and T. Chai, "Output-feedback adaptive optimal control of interconnected systems based on robust adaptive dynamic programming," *Automatica*, vol. 72, pp. 37–45, Oct. 2016.

[6] D. Liu, Q. Wei, D. Wang, X. Yang, and H. Li, *Adaptive Dynamic Programming With Applications in Optimal Control*. Cham, Switzerland: Springer, 2017.

[7] D. Liu, Y. Xu, Q. Wei, and X. Liu, "Residential energy scheduling for variable weather solar energy based on adaptive dynamic programming," *IEEE-CAA J. Automatica Sinica*, vol. 5, no. 1, pp. 36–46, Jan. 2018.

[8] Q. Wei, D. Liu, Y. Liu, and R. Song, "Optimal constrained self-learning battery sequential management in microgrid via adaptive dynamic programming," *IEEE-CAA J. Automatica Sinica*, vol. 4, no. 2, pp. 168–176, Apr. 2017.

[9] S. P. Banks, "Infinite-dimensional carleman linearization, the lie series and optimal control of nonlinear partial differential equations," *Int. J. Syst. Sci.*, vol. 23, no. 5, pp. 663–675, Apr. 1992.

[10] S. P. Banks and K. Dinesh, "Approximate optimal control and stability of nonlinear finite- and infinite-dimensional systems," *Ann. Oper. Res.*, vol. 98, no. 1/4, pp. 19–44, Dec. 2000.

[11] S. P. Banks, "Nonlinear delay systems, lie algebras and Lyapunov transformations," *IMA J. Math. Control Inform.*, vol. 19, no. 1/2, pp. 59–72, Mar. 2002.

[12] T. Cimen and S. P. Banks, "Nonlinear optimal tracking control with application to super-tankers for autopilot design," *Automatica*, vol. 40, no. 11, pp. 1845–1863, Nov. 2004.

[13] T. Cimen and S. P. Banks, "Global optimal feedback control for general nonlinear systems with nonquadratic performance criteria," *Syst. Control Lett.*, vol. 53, no. 5, pp. 327–346, Dec. 2004.

[14] M. Athans and P. L. Falb, *Optimal Control*. New York, NY, USA: McGraw-Hill, 1966.

[15] F. L. Lewis, D. Vrabie, and V. L. Syrmos, *Optimal Control*. New York, NY, USA: Wiley, 2012.

[16] G. H. Golub and C. F. Van Loan, *Matrix Computations* (2nd ed.), Baltimore, MD: Johns Hopkins University Press, 1989.

[17] R. Beard, "Improving the closed-loop performance of nonlinear systems," Ph.D. dissertation, Electrical, Computer, and Systems Engineering Department, Rensselaer Polytechnic Institute, Troy, NY, USA: 1995.

[18] M. Kelly, "An introduction to trajectory optimization: How to do your own direct collocation," *SIAM Rev.*, vol. 59, no. 4, pp. 849–904, Dec. 2017.

[19] D. P. Bertsekas, *Nonlinear Programming*, *2nd ed.*Belmont, MA, USA: Athena Scientific, 1999.

Authorized licensed use limited to: ShanghaiTech University. Downloaded on June 04,2024 at 07:12:00 UTC from IEEE Xplore. Restrictions apply.