Li *et al. Complex Eng Syst* 2023;3:20
DOI: 10.20517/ces.2023.28

Complex Engineering Systems

Research Article

Open Access

Multi-step policy evaluation for adaptive-critic-based tracking control towards nonlinear systems

Xin Li[1,2,3,4], **Jin Ren**[1,2,3,4], **Ding Wang**[1,2,3,4]

[1]Faculty of Information Technology, Beijing University of Technology, Beijing 100124, China.
[2]Beijing Key Laboratory of Computational Intelligence and Intelligent System, Beijing University of Technology, Beijing 100124, China.
[3]Beijing Laboratory of Smart Environmental Protection, Beijing University of Technology, Beijing 100124, China.
[4]Beijing Institute of Artificial Intelligence, Beijing University of Technology, Beijing 100124, China.

Correspondence to: Prof. Ding Wang, Faculty of Information Technology, Beijing University of Technology, No.100, Pingleyuan, Chaoyang District, Beijing 100124, China. E-mail: dingwang@bjut.edu.cn

How to cite this article: Li X, Ren J, Wang D. Multi-step policy evaluation for adaptive-critic-based tracking control towards nonlinear systems. *Complex Eng Syst* 2023;3:20. http://dx.doi.org/10.20517/ces.2023.28

Received: 15 Sep 2023 **First Decision:** 10 Oct 2023 **Revised:** 7 Nov 2023 **Accepted:** 8 Nov 2023 **Published:** 24 Nov 2023

Academic Editor: Hamid Reza Karimi **Copy Editor:** Dong-Li Li **Production Editor:** Dong-Li Li

Abstract

Currently, there are a large number of tracking problems in the industry concerning nonlinear systems with unknown dynamics. In order to obtain the optimal control policy, a multi-step adaptive critic tracking control (MsACTC) algorithm is developed in this paper. By constructing a steady control law, the tracking problem is transformed into a regulation problem. The MsACTC algorithm has an adjustable convergence rate during the iterative process by incorporating a multi-step policy evaluation mechanism. The convergence proof of the algorithm is provided. In order to implement the algorithm, three neural networks are built, including the model network, the critic network, and the action network. Finally, two numerical simulation examples are given to verify the effectiveness of the algorithm. Simulation results show that the MsACTC algorithm has satisfactory performance in terms of the applicability, tracking accuracy, and convergence speed.

Keywords: Adaptive critic control, multi-step policy evaluation, nonlinear systems, optimal tracking control

1. INTRODUCTION

In practical engineering applications, the controller design should not only meet the basic performance requirements but also need to further improve the control effect and reduce costs[1–4]. However, real industrial systems are often complex nonlinear systems[5,6]. Solving the Hamilton-Jacobi-Bellman (HJB) equations is an unavoidable obstacle when designing optimal control policies for nonlinear systems[7]. Considering that the analytical solution of the HJB equation is difficult to obtain, adaptive dynamic programming (ADP) based on the actor-critic framework approximates the solution of the HJB equation by the iteration[8,9]. The ADP algorithm exhibits strong adaptive and optimization capabilities by incorporating the advantages of reinforcement learning, neural networks, and dynamic programming[10,11]. Therefore, the ADP algorithm has a satisfactory performance in solving the HJB equation. Through continuous development, the ADP algorithm has become one of the key methods for solving optimal control problems of nonlinear systems[12–14]. In terms of the iterative form, the ADP algorithms can be categorized into value iteration (VI) and policy iteration (PI)[15,16]. PI has faster convergence but requires an initial admissible control policy[17]. However, it is difficult to obtain an initial admissible control policy for nonlinear systems. VI does not require an initial admissible control policy but has a slower convergence rate[18].

In the industry, we often need to solve more complex tracking problems for nonlinear systems[19–24], including the consensus tracking problem for multi-agent systems[22] and the output tracking control for single-agent systems[25]. In fact, the regulation problem can be seen as a simplified special case of the tracking problem. To overcome these challenges, some scholars have started to develop optimal learning control algorithms based on the ADP framework for solving tracking problems[26]. Currently, tracking control algorithms based on the ADP framework can be categorized into three groups. The first class of tracking control algorithms constructs the augmented system with respect to the original system and the reference trajectory and then builds the cost function based on the state vector and the control input of the augmented system[27,28]. Although this type of tracking control algorithm has a well-established theory, it cannot completely eliminate tracking errors. The second class of tracking control algorithms utilizes the square of the tracking error at the next moment to build a novel utility function[19]. With the novel utility function, the tracking error can be completely eliminated. However, the second class of tracking control algorithms is currently only applicable to affine nonlinear systems with known system models, which greatly reduces the application value of the algorithm. The third type of tracking control algorithm transforms the tracking problem into the regulation problem by constructing a virtual steady control[29]. Thanks to the steady control, the third type of tracking control algorithm also completely eliminates tracking errors and is suitable for nonlinear systems. It should be emphasized that solving the steady control imposes a large computational burden on the algorithm.

Therefore, we expected that a speed-up mechanism could be designed for the tracking control algorithm with steady control to improve the convergence speed. Inspired by the idea of eligibility traces, Al-Dabooni *et al*. designed an online n-step ADP algorithm with higher learning efficiency[30]. Ha *et al*. introduced a relaxation factor in the framework of the offline ADP algorithm[31]. By adjusting the value of the relaxation factor, the convergence rate of the cost function could also be increased or decreased. Zhao *et al*. designed an incremental ADP algorithm by constructing a new type of cost function, which showed higher learning efficiency in solving zero-sum game problems for nonlinear systems[29]. Luo *et al*. successfully combined the advantages of VI and PI by designing a multi-step policy evaluation mechanism[17,25,32]. The convergence speed of the algorithm was greatly improved by increasing the policy evaluation step, which did not require admissible control policies.

In this context, a multi-step adaptive critic tracking control (MsACTC) algorithm is established by introducing a multi-step policy evaluation mechanism into the steady tracking control algorithm. Compared to the general tracking control algorithm with steady control, the MsACTC algorithm exhibits higher learning efficiency as the evaluation step increases. Furthermore, the convergence proof of the MsACTC algorithm is given. The MsACTC algorithm is successfully implemented through neural networks and the least square method. Finally,

Table 1. Summary of mathematical symbols

$\mathbb{R}^s$	$\|\|A\|\|$	I_s
The set of all s-dimensional real vectors	The norm of vector A	The $s \times s$ identity matrix
$\mathbb{R}^{s\times m}$	$\|c\|$	T
The set of all $s \times m$ real matrices	The absolute value of c	The transpose symbol
$\mathbb{N}$	$\mathbb{N}^+$	Ω
The set of all non-negative integers	The set of all positive integers	A compact subset of $\mathbb{R}^s$

we verify the optimization ability and learning efficiency of the proposed algorithm through simulations of two nonlinear systems. Table 1 explains the meaning of the mathematical symbols used in this article.

2. PROBLEM DESCRIPTION

Consider such a class of discrete-time nonlinear systems

$$x_{k+1} = F\left(x_k, u(x_k)\right), \tag{1}$$

where $x_k \in \mathbb{R}^s$ represents the state vector, and $u(x_k) \in \mathbb{R}^m$ represents the control input. $F(\cdot,\cdot)$ is the unknown system function. In addition, we propose two assumptions to describe equation (1) more specifically.

Assumption 1 *$x(k)$ and $u(x_k)$ are observable, and there exists a continuous control policy on Ω such that the system (1) is asymptotically stable.*

Assumption 2 *$F(\cdot,\cdot)$ is smooth differentiable with respect to its arguments.*

The reference trajectory can be denoted as

$$r_{k+1} = H(r_k), \tag{2}$$

where $H(\cdot)$ denotes the known trajectory function. The tracking error e_k is defined as

$$e_k = x_k - r_k. \tag{3}$$

We assume that there exists a steady control $v(r_k)$ satisfying the following equation:

$$r_{k+1} = F\left(r_k, v(r_k)\right). \tag{4}$$

The tracking control policy $\mu(e_k)$ is denoted as

$$\mu(e_k) = u(x_k) - v(r_k). \tag{5}$$

According to equations (1)-(5), we can obtain the following error system:

$$\begin{aligned} e_{k+1} &= F\left(x_k, u(x_k)\right) - H(r_k) \\ &= F\left(e_k + r_k, \mu(e_k) + v(r_k)\right) - H(r_k) \\ &= \Phi(e_k, \mu(e_k)). \end{aligned} \tag{6}$$

The cost function is defined as

$$\begin{aligned} V_k &= \sum_{z=k}^{\infty} U\left(e_z, \mu(e_z)\right) \\ &= U\left(e_k, \mu(e_k)\right) + \sum_{z=k+1}^{\infty} U\left(e_z, \mu(e_z)\right) \\ &= U\left(e_k, \mu(e_k)\right) + V_{k+1}, \end{aligned} \tag{7}$$

where $U(e_k, \mu(e_k)) = x_k^{\mathsf{T}} Q x_k + \mu^{\mathsf{T}}(e_k) R \mu(e_k)$ is the utility function. Q and R are positive definite matrices. We need to minimize the cost function by continuously adjusting the tracking control policy. The optimal cost function and the optimal control policy are expressed as follows:

$$V_k^* = \min_{\mu(e_k)} \left\{ U(e_k, \mu(e_k)) + V_{k+1}^* \right\}, \tag{8}$$

and

$$\mu^*(e_k) = \arg\min_{\mu(e_k)} \left\{ U(e_k, \mu(e_k)) + V_{k+1}^* \right\}. \tag{9}$$

So far, the tracking problem of the system (1) has been successfully transformed into the regulation problem of the system (6).

3. MULTI-STEP ADAPTIVE CRITIC TRACKING CONTROL ALGORITHM

It should be noted that equation (8) is the HJB equation, and its analytical solution is difficult to obtain directly. In addition, since the system function $F(\cdot)$ is unknown, the steady control cannot be directly obtained by solving equation (3) either. Therefore, the MsACTC algorithm is designed to overcome these challenges.

3.1 Algorithm design

According to equation (6), we have

$$\begin{aligned} V_k &= U(e_k, \mu(e_k)) + V_{k+1} \\ &= U(e_k, \mu(e_k)) + U(e_{k+1}, \mu(e_{k+1})) + V_{k+2} \\ &= \sum_{z=k}^{k+n-1} U(e_z, \mu(e_z)) + V_{k+n}, \end{aligned} \tag{10}$$

where $n \in \mathbb{N}^+$ denotes the step size of policy evaluation. To integrate the convenience of VI and the efficiency of PI, we introduce the multi-step policy evaluation mechanism. Based on this mechanism, the MsACTC algorithm has a faster convergence rate without the need for an admissible control policy.

Construct the sequence of iterative cost functions $\left\{ V_k^{(i)} \right\}$ and the sequence of iterative tracking control policies $\left\{ \mu^{(i)}(e_k) \right\}$, where $i \in \mathbb{N}$ is the iteration index. Define the initial cost function $V_k^{(0)} = e_k^{\mathsf{T}} P e_k$, where P is a positive definite matrix. Then, the policy improvement is represented as follows:

$$\mu^{(i)}(e_k) = -\frac{1}{2} R^{-1} \left(\frac{\partial e_{k+1}}{\partial \mu(e_k)} \right)^{\mathsf{T}} \frac{\partial V_{k+1}^{(i)}}{\partial e_{k+1}}. \tag{11}$$

The policy evaluation is expressed as follows:

$$V_k^{(i+1)} = \sum_{z=k}^{k+n-1} U\left(e_z, \mu^{(i)}(e_z) \right) + V_{k+n}^{(i)}. \tag{12}$$

By iterating continuously between equations (11) and (12), the cost function and the control policy converge to their optimal values as i tends to infinity, respectively. In practice, it is not possible for the algorithm to perform an infinite number of iterations. Therefore, we set the following stopping criterion:

$$\left| V_k^{(i+1)} - V_k^{(i)} \right| \leq \eta, \tag{13}$$

where η is a positive constant. When inequation (13) is satisfied, the iteration stops. In this case, the cost function and the control policy are considered to be approximately optimal.

3.2 Theoretical analysis

To ensure that the cost function $V_k^{(i)}$ and the control policy $\mu^{(i)}(e_k)$ can converge to their optimal values, we formulate and prove Theorem 1.

Theorem 1 *If the sequences* $\{\mu^{(i)}(e_k)\}$ *and* $\left\{V_k^{(i)}\right\}$ *are updated according to equations (11) and (12), respectively, and the following condition is satisfied:*

$$V_k^{(0)} \geq \min_{\mu(e_k)} \left\{U\left(e_k, \mu(e_k)\right) + V_{k+1}^{(0)}\right\}, \tag{14}$$

then the following conclusions hold:

1) $\forall i,\ V_k^{(i+1)} \leq \min_{\mu(e_k)} \left\{U\left(e_k, \mu(e_k)\right) + V_{k+1}^{(i)}\right\} \leq V_k^{(i)}$

2) $\lim_{i\to\infty} V_k^{(i)} = V_k^*$ *and* $\lim_{i\to\infty} \mu^{(i)}(e_k) = \mu^*(e_k)$.

Proof. First, we prove the conclusion 1). When (14) holds, we have

$$\begin{aligned}
V_k^{(1)} &= \sum_{z=k}^{k+n-1} U\left(e_z, \mu^{(0)}(e_z)\right) + V_{k+n}^{(0)} \\
&= \sum_{z=k}^{k+n-2} U\left(e_z, \mu^{(0)}(e_z)\right) + U\left(e_{k+n-1}, \mu^{(0)}(e_{k+n-1})\right) + V_{k+n}^{(0)} \\
&= \sum_{z=k}^{k+n-2} U\left(e_z, \mu^{(0)}(e_z)\right) + \min_{\mu(e_k)} \left\{U\left(e_{k+n-1}, \mu(e_{k+n-1})\right) + V_{k+n}^{(0)}\right\} \\
&\leq \sum_{z=k}^{k+n-2} U\left(e_z, \mu^{(0)}(e_z)\right) + V_{k+n-1}^{(0)} \\
&\vdots \\
&\leq U\left(e_k, \mu^{(0)}(e_k)\right) + V_{k+1}^{(0)} = \min_{\mu(e_k)} \left\{U\left(e_k, \mu(e_k)\right) + V_{k+1}^{(0)}\right\}.
\end{aligned} \tag{15}$$

So far, we have shown that the conclusion 1) holds for $i = 0$. Next, we assume that the conclusion 1) holds for the iteration index $i - 1$, which means that

$$V_k^{(i)} \leq \min_{\mu(e_k)} \left\{U\left(e_k, \mu(e_k)\right) + V_{k+1}^{(i-1)}\right\} \leq V_k^{(i-1)}. \tag{16}$$

By further derivation, we get

$$\begin{aligned}
V_k^{(i)} &= \sum_{z=k}^{k+n-1} U\left(e_z, \mu^{(i-1)}(e_z)\right) + V_{k+n}^{(i-1)} \\
&\geq \sum_{z=k}^{k+n-1} U\left(e_z, \mu^{(i-1)}(e_z)\right) + \min_{\mu(e_k)} \left\{U\left(e_{k+n}, \mu(e_{k+n})\right) + V_{k+n+1}^{(i-1)}\right\} \\
&= \sum_{z=k}^{k+n} U\left(e_z, \mu^{(i-1)}(e_z)\right) + V_{k+n+1}^{(i-1)} \\
&= U\left(e_k, \mu^{(i-1)}(e_k)\right) + V_{k+1}^{(i)} \\
&\geq \min_{\mu(e_k)} \left\{U\left(e_k, \mu(e_k)\right) + V_{k+1}^{(i)}\right\}.
\end{aligned} \tag{17}$$

Based on inequations (16) and (17), we have

$$\begin{aligned} V_k^{(i+1)} &= \sum_{z=k}^{k+n-1} U\left(e_z, \mu^{(i)}(e_z)\right) + V_{k+n}^{(i)} \\ &= \sum_{z=k}^{k+n-2} U\left(e_z, \mu^{(i)}(e_z)\right) + U\left(e_{k+n-1}, \mu^{(i)}(e_{k+n-1})\right) + V_{k+n}^{(i)} \\ &= \sum_{z=k}^{k+n-2} U\left(e_z, \mu^{(i)}(e_z)\right) + \min_{\mu(e_k)} \left\{U\left(e_{k+n-1}, \mu(e_{k+n-1})\right) + V_{k+n}^{(i)}\right\} \\ &\leq \sum_{z=k}^{k+n-2} U\left(e_z, \mu^{(i)}(e_z)\right) + V_{k+n-1}^{(i)} \\ &\vdots \\ &\leq U\left(e_k, \mu^{(i)}(e_k)\right) + V_{k+1}^{(i)} = \min_{\mu(e_k)} \left\{U\left(e_k, \mu(e_k)\right) + V_{k+1}^{(i)}\right\}. \end{aligned} \tag{18}$$

According to mathematical induction, the proof of the conclusion 1) is completed.

Next, we prove the conclusion 2). Since the sequence of cost functions $\left\{V_k^{(i)}\right\}$ is a monotonically nonincreasing sequence and $V_k^{(i)} \geq 0$, the limit of the sequence exists. Therefore, we can get

$$V_k^{(\infty)} \leq \min_{\mu(e_k)} \left\{U\left(e_k, \mu(e_k)\right) + V_{k+1}^{(\infty)}\right\} \leq V_k^{(\infty)}, \tag{19}$$

which means

$$V_k^{(\infty)} = \min_{\mu(e_k)} \left\{U\left(e_k, \mu(e_k)\right) + V_{k+1}^{(\infty)}\right\}. \tag{20}$$

By observation, equations (20) and (8) are equivalent. This means that $\lim_{i\to\infty} V_k^{(i)} = V_k^*$. According to equation (9), the corresponding tracking control policy also satisfies $\lim_{i\to\infty} \mu^{(i)}(e_k) = \mu^*(e_k)$. The proof of Theorem 1 is completed.

Theorem 2 *Based on equations (1)-(6), $\partial e_{k+1}/\partial \mu(e_k) = \partial x_{k+1}/\partial u(x_k)$ holds.*

Proof. According to the error system (6), we have

$$\begin{aligned} \frac{\partial e_{k+1}}{\partial \mu(e_k)} &= \frac{\partial \left(F\left(e_k + r_k, \mu(e_k) + v(r_k)\right) - H(r_k)\right)}{\partial \mu(e_k)} \\ &= \frac{\partial F\left(e_k + r_k, \mu(e_k) + v(r_k)\right)}{\partial \left(\mu(e_k) + v(r_k)\right)}. \end{aligned} \tag{21}$$

Besides, we have

$$\begin{aligned} \frac{\partial x_{k+1}}{\partial u(x_k)} &= \frac{\partial F\left(x_k, u(x_k)\right)}{\partial u(x_k)} \\ &= \frac{\partial F\left(e_k + r_k, \mu(e_k) + v(r_k)\right)}{\partial \left(\mu(e_k) + v(r_k)\right)}. \end{aligned} \tag{22}$$

The proof of Theorem 2 is completed. In this section, Theorem 1 is successfully proved by utilizing mathematical induction, and it provides a theoretical justification for the convergence of the MsACTC algorithm. Theorem 2 gives an equivalence between $\partial e_{k+1}/\partial \mu(e_k)$ and $\partial x_{k+1}/\partial u(x_k)$. According to Theorem 2, the implementation of the algorithm can be simplified.

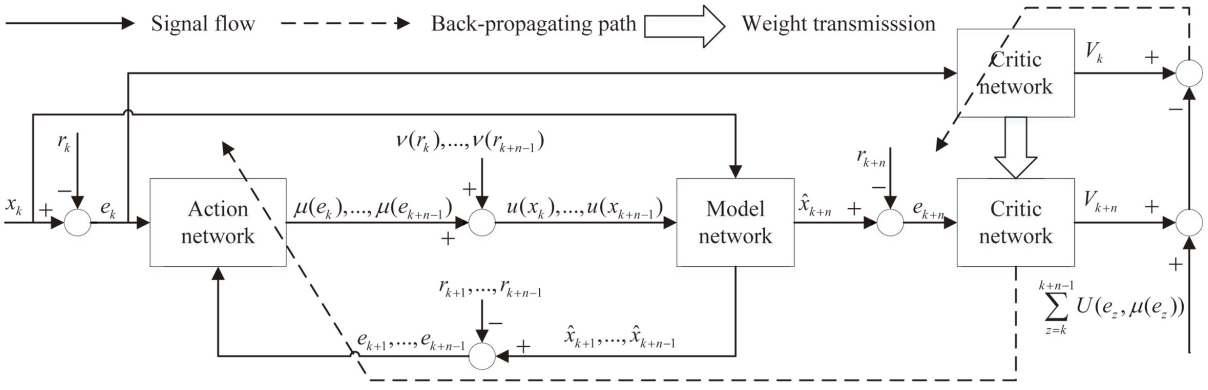

Figure 1. The structure diagram of the MsACTC algorithm. MsACTC: Multi-step adaptive critic tracking control.

Remark 1 *To ensure that the condition (14) holds, $V_k^{(0)}$ must be a positive definite function. It should be emphasized that the condition (14) is only a sufficient condition instead of a sufficiently necessary condition for the convergence of the cost function. In fact, even if the cost function is initialized as $V_k^{(0)} = 0$, the cost function still converges to the optimal value. This is verified by the simulation in Section 5, which shows that the convergence of the algorithm deserves to be further investigated.*

4. IMPLEMENTATION OF ALGORITHM

To implement the algorithm, three neural networks are constructed, including a model network (MN), a critic network (CN), and an action network (AN). Figure 1 illustrates the overall structure of the MsACTC algorithm. It should be noted that the training of the MN is carried out independently. The training of the CN and the AN need to cooperate with each other iteratively. It should be noted that the implementation approach used in this paper is similar to that in [22].

The MN is used to estimate the dynamics of the system (1) and to solve the steady control $v(r_k)$. Since the MN is required for the training process of the CN and the AN, we should first complete the training of the MN. The output of the MN can be expressed as

$$\hat{x}_{k+1} = w_2^{(j)\mathsf{T}} \sigma\left(w_1^{(j)\mathsf{T}} \bar{x}_k + \beta_1^{(j)}\right) + \beta_2^{(j)}, \tag{23}$$

where $\bar{x}_k = \left[x_k^{\mathsf{T}}, u^{\mathsf{T}}(x_k)\right]^{\mathsf{T}}$. $w_1^{(j)} \in \mathbb{R}^{(s+m)\times l_m}$ and $w_2^{(j)} \in \mathbb{R}^{l_m \times s}$ denote the weight vectors. $\beta_1^{(j)} \in \mathbb{R}^{l_m}$ and $\beta_2^{(j)} \in \mathbb{R}^s$ are bias vectors. l_m denotes the number of neurons in the hidden layer for the MN. j denotes the training round index.

In order to make the MN with higher recognition accuracy, we utilize the neural network toolbox in Matlab to train the MN. The training parameter settings are given in Section 5.

After the MN is trained, according to equation (4), we can obtain the steady control $v(r_k)$ by solving the following equation:

$$r_{k+1} = w_2^{\mathsf{T}} \sigma\left(w_1^{\mathsf{T}} \left[r_k^{\mathsf{T}}, v^{\mathsf{T}}(r_k)\right]^{\mathsf{T}} + \beta_1^{(j)}\right) + \beta_2^{(j)}. \tag{24}$$

The CN is used to approximate the cost function, and its output can be expressed as

$$\hat{V}_k^{(i)} = \phi^{\mathsf{T}}(e_k)\omega^{(i)}, \tag{25}$$

where $\omega^{(i)} \in \mathbb{R}^{l_c}$. The activation function $\phi(e_k) = [\phi_1(e_k), \cdots, \phi_{l_c}(e_k)]^{\mathsf{T}}$. l_c denotes the number of neurons in the hidden layer for the CN. According to equations (12) and (25), we have

$$\phi^{\mathsf{T}}(e_k)\omega^{(i+1)} = \sum_{z=k}^{k+n-1} U\left(e_z, \mu^{(i)}(e_z)\right) + \phi^{\mathsf{T}}(e_{k+n})\omega^{(i)}. \tag{26}$$

A sample set $\psi_M = \left\{e^{[q]} | e_k^{[q]} \in \Omega_e, q = 1, 2 \cdots, M\right\}$ is collected on Ω, where M denotes the maximum number of samples. Based on the MN and $\mu^{(i)}(e_k)$, we can predict the future error vector $e_{k+n}^{[q]}$. Considering the sample set ψ_M, equation (26) can be rewritten as

$$\phi^{\mathsf{T}}\left(e_k^{[q]}\right)\omega^{(i+1)} = \zeta^{[q](i)}, q = 1, 2, \ldots, M, \tag{27}$$

where $\zeta^{[q](i)} = \sum_{z=k}^{k+n-1} U\left(e_z^{[q]}, \mu^{(i)}\left(e_z^{[q]}\right)\right) + \phi^{\mathsf{T}}\left(e_{k+n}^{[q]}\right)\omega^{(i)}$. Using the least-square method, $\omega^{(i+1)}$ can be calculated as

$$\omega^{(i+1)} = (\lambda^{\mathsf{T}}\lambda)^{-1}\lambda^{\mathsf{T}}\zeta^{(i)}, \tag{28}$$

where $\lambda = \left[\phi^{\mathsf{T}}\left(e_k^{[1]}\right), \phi^{\mathsf{T}}\left(e_k^{[2]}\right), \cdots, \phi^{\mathsf{T}}\left(e_k^{[M]}\right)\right]^{\mathsf{T}}$ and $\zeta^{(i)} = \left[\zeta^{[1](i)}, \zeta^{[2](i)}, \cdots, \zeta^{[M](i)}\right]^{\mathsf{T}}$.

The AN is used to approximate the tracking control policy, and its output can be expressed as

$$\hat{\mu}^{(i)}(e_k) = \varphi^{\mathsf{T}}(e_k)\varpi^{(i)}, \tag{29}$$

where $\varpi^{(i)} \in \mathbb{R}^{l_a}$. The activation function $\varphi(e_k) = [\varphi_1(e_k), \cdots, \varphi_{l_a}(e_k)]^{\mathsf{T}}$. l_a denotes the number of neurons in the hidden layer for the AN. According to equations (11) and (29), we have

$$\hat{\mu}^{(i+1)}(e_k) = \varphi^{\mathsf{T}}(e_k)\varpi^{(i+1)} = -\frac{1}{2}R^{-1}\left(\frac{\partial \hat{x}_{k+1}}{\partial u^{(i)}(x_k)}\right)^{\mathsf{T}} \frac{\partial \hat{V}_{k+1}^{(i+1)}}{\partial u^{(i)}(e_{k+1})}. \tag{30}$$

Considering the sample set ψ_M, equation (30) can be rewritten as

$$\varphi^{\mathsf{T}}\left(e_k^{[q]}\right)\varpi^{(i+1)} = \theta^{[q](i)}, q = 1, 2 \cdots, M, \tag{31}$$

where $\theta^{[q](i)} = -\frac{1}{2}R^{-1}\left(\frac{\partial \hat{x}_{k+1}^{[q]}}{\partial u^{(i)}(x_k^{[q]})}\right)^{\mathsf{T}} \frac{\partial \hat{V}_{k+1}^{(i+1)}}{\partial u^{(i)}(\hat{e}_{k+1}^{[q]})}$. Thus, $\varpi^{(i+1)}$ can be calculated as

$$\varpi^{(i+1)} = (\rho^{\mathsf{T}}\rho)^{-1}\rho^{\mathsf{T}}\theta^{(i)}, \tag{32}$$

where $\rho = \left[\varphi^{\mathsf{T}}\left(e_k^{[1]}\right), \varphi^{\mathsf{T}}\left(e_k^{[2]}\right), \cdots, \varphi^{\mathsf{T}}\left(e_k^{[M]}\right)\right]^{\mathsf{T}}$ and $\theta^{(i)} = \left[\theta^{[1](i)}, \theta^{[2](i)}, \cdots, \theta^{[M](i)}\right]^{\mathsf{T}}$.

5. SIMULATION RESULT

In this section, we complete two simulations. Example 1 is to illustrate the effectiveness of the MsACTC algorithm and the correctness of Theorem 1. Example 2 is to show that the condition (14) is not sufficiently necessary for the convergence of the cost function.

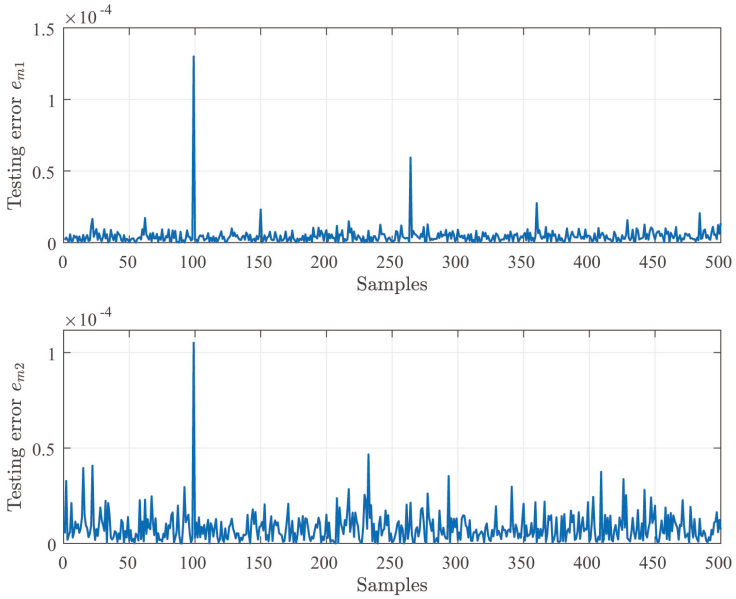

Figure 2. Testing errors of the MN (Example 1). MN: Model network.

5.1 Example 1

Consider the modified Van der Pol's oscillator system

$$x_{k+1} = \begin{bmatrix} x_{k1} + 0.1x_{k2} \\ -0.1x_{k1} + 1.1x_{k2}^2 - 0.1x_{k1}^2 x_{k2} \end{bmatrix} + \begin{bmatrix} 0.35 & 0 \\ 0 & 0.35 \end{bmatrix} u(x_k). \tag{33}$$

The reference trajectory is defined as

$$r_{k+1} = \begin{bmatrix} 1 & 0.05 \\ -0.05 & 1 \end{bmatrix} r_k. \tag{34}$$

Before the neural networks are trained, we make a sample set of x_k in the region where $-1 \le x_{k1} \le 1$ and $-1 \le x_{k2} \le 1$. In order to obtain the sample set of error vectors e_k, the sample set of the reference trajectory r_k is also constructed in the region where $-1 \le r_{k1} \le 1$ and $-1 \le r_{k2} \le 1$. Assuming that the system (33) is unknown, we need to construct the MN using the neural network toolbox and the data generated by the system (33). The parameters of the toolbox are set as follows: the activation function is chosen as $\sigma(z) = (e^z - e^{-z})/(e^z + e^{-z})$, the optimization algorithm is chosen as Levenberg-Marquardt backpropagation, the maximum training number of the MN is set as 1000, and the loss function is chosen as the mean squared normalized error function. The other parameters in the MsACTC algorithm are set as $Q = I_2$, $R = 5I_2$, $P = 6I_2$, $\eta = 10^{-7}$, $M = 961$, $l_m = 20$, $l_a = 5$, and $l_c = 7$. Figure 2 shows the testing error after the training of the MN, and the result is satisfactory.

We construct the activation functions for the CN and the AN as

$$\begin{cases} \phi(e_k) = \left[e_{k1}^2, e_{k2}^2, e_{k1}e_{k2}, e_{k1}^2 e_{k2}, e_{k1}e_{k2}^2, e_{k1}^3, e_{k2}^3\right]^{\mathsf{T}} \\ \varphi(e_k) = \left[e_{k1}, e_{k2}, e_{k1}e_{k2}, e_{k1}^2, e_{k2}^2\right]^{\mathsf{T}} \end{cases} \tag{35}$$

In order to satisfy the condition (14), the weight vector $\omega^{(0)}$ of the CN is initialized by one pre-training. $\omega^{(0)}$ satisfies $\hat{V}_k^{(0)} = \phi^{\mathsf{T}}(e_k)\omega^{(0)} = e_k^{\mathsf{T}} P e_k$. The weight matrix of the AN is initialized as zero. Before starting the iteration, we set the maximum iteration number $i_{\mathrm{M}} = 25$. We use the different step sizes n of policy evaluation for the simulation to verify the effect of n on the convergence speed of the MsACTC algorithm.

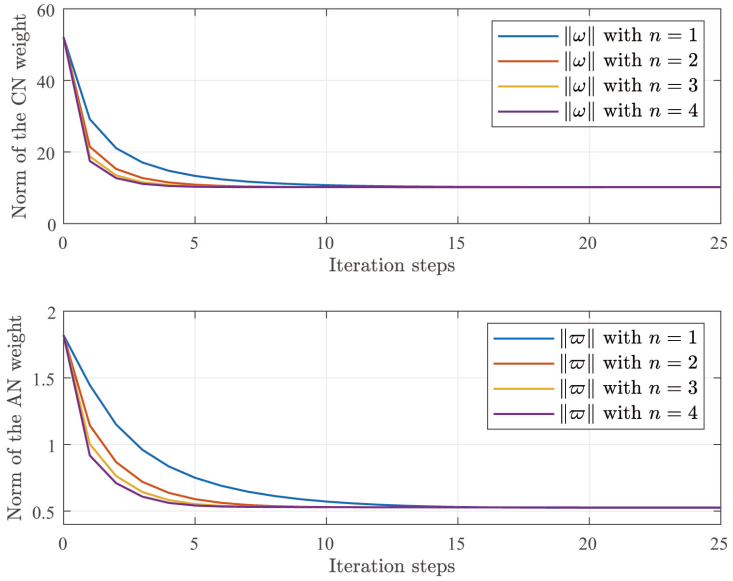

Figure 3. Convergence processes of network weights with different values of n (Example 1). AN: Action network; CN: critic network.

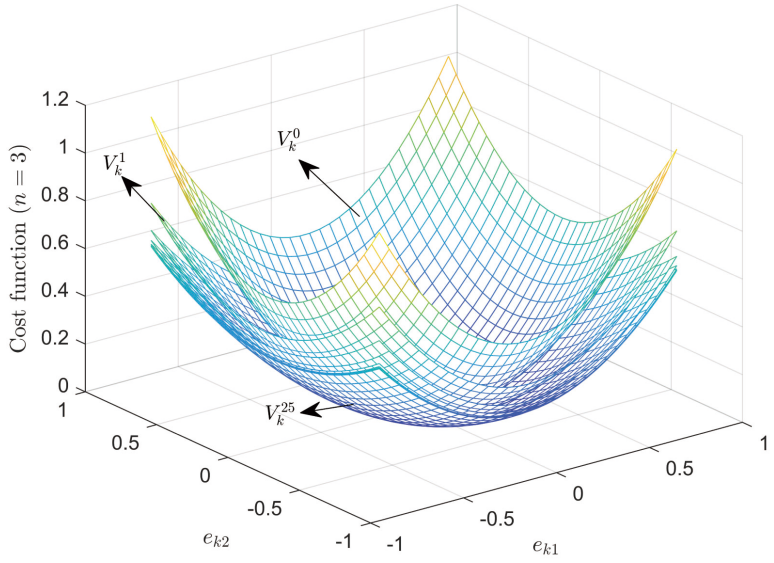

Figure 4. Convergence processes of the cost function for $n = 3$ (Example 1).

In order to compare the convergence speeds under different values of n, Figure 3 utilizes the convergence process of the norm of the weights to reflect the iterative process of the cost function and the tracking control policy. According to Figure 3, we can observe that the larger the value of n, the faster the weights converge. Moreover, the weights corresponding to different values of n eventually converge to the same value, which indicates the optimality of the MsACTC algorithm. Figure 4 shows the iterative process of the cost function for $n = 3$ through a mesh plot.

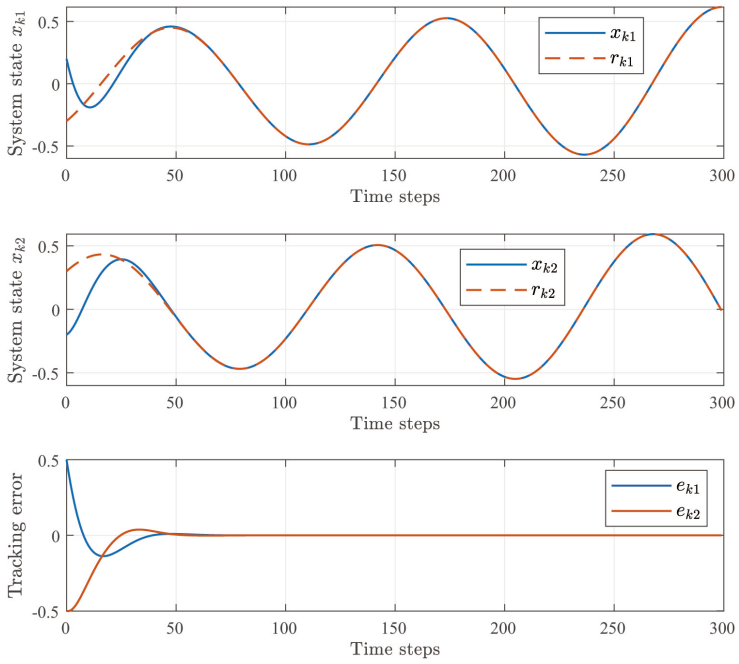

Figure 5. Trajectories of state vectors and tracking errors (Example 1).

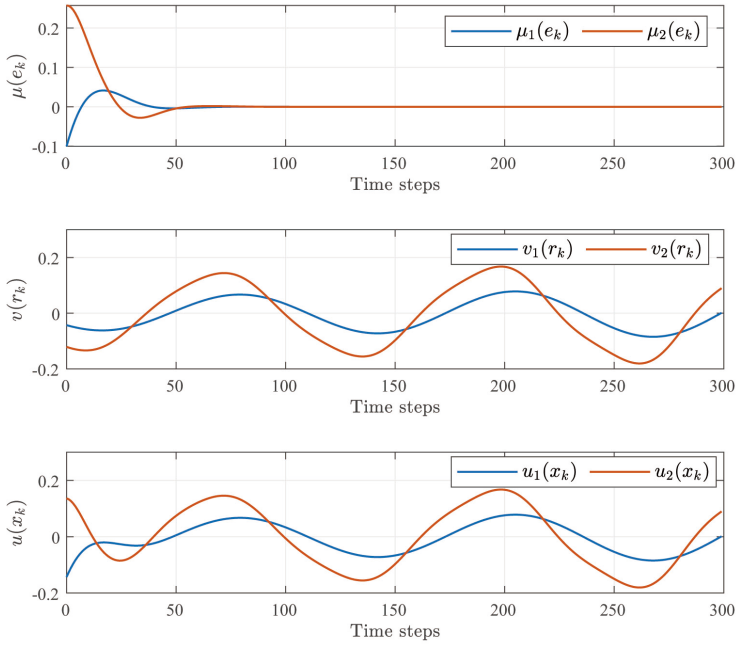

Figure 6. Trajectories of control inputs (Example 1).

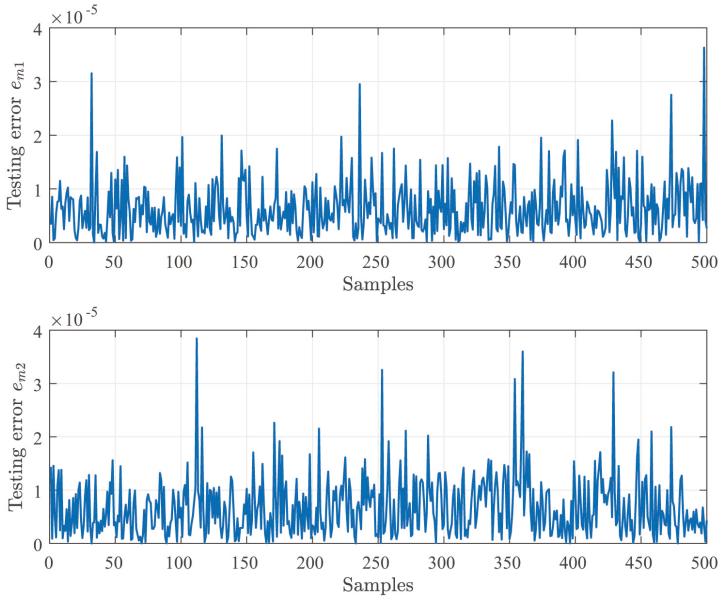

Figure 7. Testing errors of the MN (Example 2). MN: Model network.

We apply the optimal control policy obtained by the MsACTC algorithm for $n = 3$ to the system (33) for 300 time steps. We set $x_0 = [0.2, -0.2]^\mathsf{T}$ and $r_0 = [-0.3, -0.3]^\mathsf{T}$. The changing processes of the system state and tracking error are shown in Figure 5. The variation of control inputs is shown in Figure 6. With the control policy, the system states track the reference trajectory after 50 time steps.

5.2 Example 2

Consider the modified torsional pendulum system

$$x_{k+1} = \begin{bmatrix} x_{k1} + 0.1x_{k2} + 0.12u_1(x_k) \\ -0.225\sin(x_{k1}) + 0.975x_{k2} + 0.125(\tanh(u_2(x_k)) + u_2(x_k)) \end{bmatrix}. \tag{36}$$

Due to the excellent adaptive ability, the MsACTC algorithm does not have strict requirements for the settings of the parameters. Therefore, the parameter values and the setting of activation functions are kept the same as those in Example 1. In Example 2, we set $V_k^{(0)} = 0$ to show that the condition (14) is not necessary for the convergence of $V_k^{(i)}$. That is, $\varpi^{(0)}$ and $\omega^{(0)}$ are set as zero. In Example 2, we set $x_0 = [0.5, -0.5]^\mathsf{T}$ and $r_0 = [-0.5, 0.3]^\mathsf{T}$. After completing the initialization, the procedure of the experiment is the same as that in Example 1. The simulation results are shown in Figures 7-11. It is worth noting the direction of convergence of weights and the cost function in Figures 8 and 9. For Example 1, the direction of convergence is illustrated in Figures 3 and 4 is from the top to the bottom. For Example 2, the direction of convergence in Figures 8 and 9 is from the bottom to the top.

Even if the condition (14) does not hold, $V_k^{(i)}$ and $\mu^{(i)}(e_k)$ can still converge to V_k^* and $\mu^*(e_k)$ through the iteration. As n increases, the algorithm converges faster. However, it is worth noting that if the condition (14) is not satisfied, too large n will make the cost function diverge during the iteration.

6. CONCLUSIONS AND OUTLOOK

In this paper, the MsACTC algorithm is developed by introducing the multi-step policy evaluation mechanism into the adaptive critic tracking control algorithm. For nonlinear systems with unknown models, the MN is

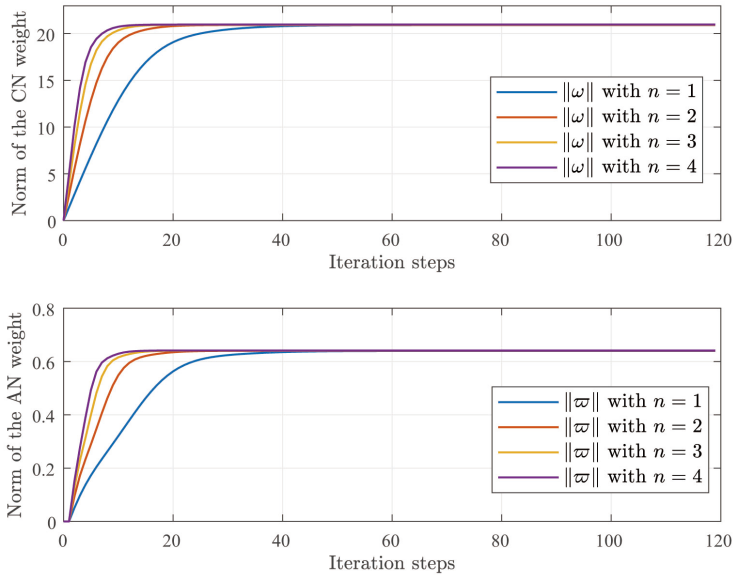

Figure 8. Convergence processes of network weights with different values of n (Example 2). AN: Action network; CN: critic network.

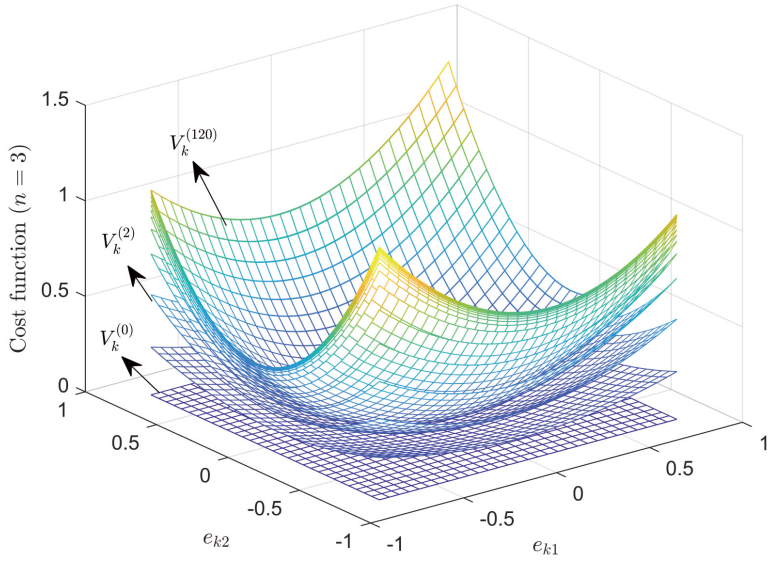

Figure 9. Convergence processes of the cost function for $n = 3$ (Example 2).

built based on data to estimate the state vector in the future time. By solving the steady control $v(r_k)$ through the MN, the tracking problem is transformed into the regulation problem. We give the convergence proof of the MsACTC algorithm. The weights of the CN and the AN are updated based on the least-square method. Simulation results verify the effectiveness of the MsACTC algorithm and the correctness of the theoretical results. It is worth exploring how to obtain the most suitable value of n in the MsACTC algorithm. When the condition (14) holds, the control policy is admissible after one policy improvement. Therefore, n can be assigned arbitrarily in the range of positive integers. However, when the value of n is large enough, even if

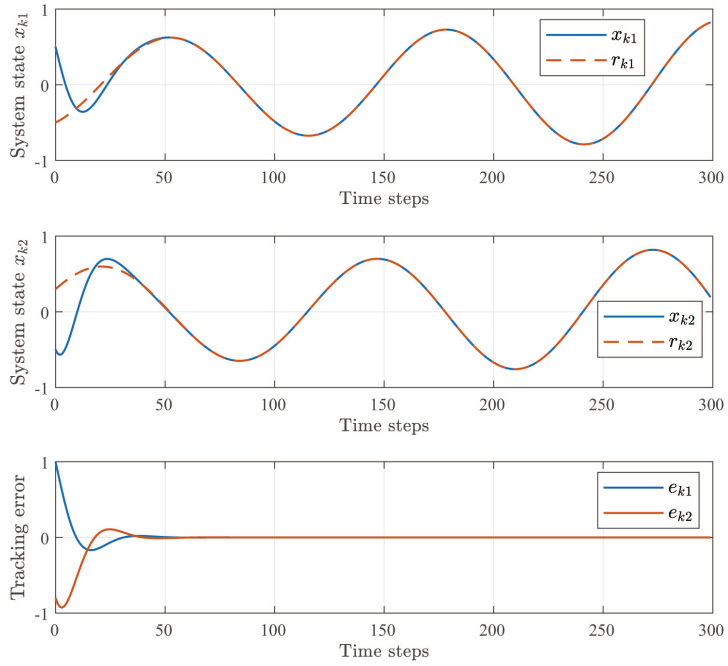

Figure 10. Trajectories of state vectors and tracking errors (Example 2).

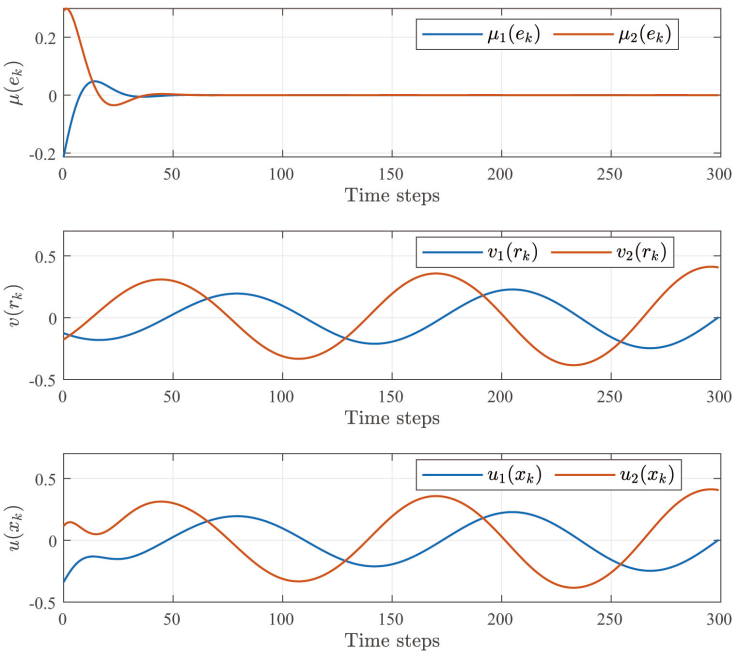

Figure 11. Trajectories of control inputs (Example 2).

n continues to increase, the convergence speed will not be significantly improved. In this case, increasing the value of n will lead to a waste of computing resources. If the condition (14) is not satisfied, then the algorithm cannot guarantee that the control policy is admissible, which means that the value of n is no longer arbitrary. This is because an unstable control policy will cause the state vector to diverge during multi-step policy evaluation. In general, a satisfactory method for selecting the value of n has not yet been proposed. In the future, we may continue to investigate this issue. In addition, this paper only uses two digital simulation examples to test the MsACTC algorithm. This means that the potential application areas of the algorithm have not been fully explored. Therefore, we will try to verify the performance of the algorithm in more application scenarios in the future.

DECLARATIONS

Authors' contributions

Formal analysis, validation, writing - original draft: Li X
Methodology, Supervision, Writing - review and editing: Ren J
Investigation, Supervision, Writing - review and editing: Wang D

Availability of data and materials

Not applicable.

Financial support and sponsorship

This work was supported in part by the National Key Research and Development Program of China under Grant 2021ZD0112302 and in part by the National Natural Science Foundation of China under Grant 62222301, Grant 61890930-5, and Grant 62021003.

Conflicts of interest

All authors declared that there are no conflicts of interest.

Ethical approval and consent to participate

Not applicable

Consent for publication

Not applicable

REFERENCES

1. Maamoun KSA, Karimi HR. Reinforcement learning-based control for offshore crane load-landing operations. *Complex Eng Syst* 2022;2:13. DOI
2. Wei Q, Liu D, Shi G, Liu Y. Multibattery optimal coordination control for home energy management systems via distributed iterative adaptive dynamic programming. *IEEE Trans Ind Electron* 2015;62:4203-14. DOI
3. Firdausiyah N, Taniguchi E, Qureshi AG. Multi-agent simulation-adaptive dynamic programming based reinforcement learning for evaluating joint delivery systems in relation to the different locations of urban consolidation centres. *Transp Res Proc* 2020;46:125-32. DOI
4. Li S, Ding L, Gao H, Liu YJ, Huang L, Deng Z. ADP-based online tracking control of partially uncertain time-delayed nonlinear system and application to wheeled mobile robots. *IEEE Trans Cybern* 2020;50:3182-94. DOI
5. Sun T, Sun XM. An adaptive dynamic programming scheme for nonlinear optimal control with unknown dynamics and its application to turbofan engines. *IEEE Trans Ind Inf* 2021;17:367-76. DOI
6. Davari M, Gao W, Jiang ZP, Lewis FL. An optimal primary frequency control based on adaptive dynamic programming for islanded modernized microgrids. *IEEE Trans Automat Sci Eng* 2021;18:1109-21. DOI
7. Zhao M, Wang D, Qiao J, Ha M, Ren J. Advanced value iteration for discrete-time intelligent critic control: a survey. *Artif Intell Rev* 2023;56:12315-46. DOI

8. Lewis FL, Vrabie D. Reinforcement learning and adaptive dynamic programming for feedback control. *IEEE Circuits Syst Mag* 2009;9:32-50. DOI
9. Liu D, Xue S, Zhao B, Luo B, Wei Q. Adaptive dynamic programming for control: a survey and recent advances. *IEEE Trans Syst Man Cybern Syst* 2021;51:142-60. DOI
10. Zhang H, Cui L, Zhang X, Luo Y. Data-driven robust approximate optimal tracking control for unknown general nonlinear systems using adaptive dynamic programming method. *IEEE Trans Neural Netw* 2011;22:2226-36. DOI
11. Wang D, Wang J, Zhao M, Xin P, Qiao J. Adaptive multi-step evaluation design with stability guarantee for discrete-time optimal learning control. *IEEE/CAA J Autom Sinica* 2023;10:1797-809. DOI
12. Wang D, Hu L, Zhao M, Qiao J. Dual event-triggered constrained control through adaptive critic for discrete-time zero-sum games. *IEEE Trans Syst Man Cybern Syst* 2023;53:1584-95. DOI
13. Fairbank M, Alonso E, Prokhorov D. Simple and fast calculation of the second-order gradients for globalized dual heuristic dynamic programming in neural networks. *IEEE Trans Neural Netw Learn Syst* 2012;23:1671-6. DOI
14. Gao W, Deng C, Jiang Y, Jiang ZP. Resilient reinforcement learning and robust output regulation under denial-of-service attacks. *Automatica* 2022;142:110366. DOI
15. Liu D, Wei Q. Policy iteration adaptive dynamic programming algorithm for discrete-time nonlinear systems. *IEEE Trans Neural Netw Learn Syst* 2013;25:621-34. DOI
16. Wei Q, Liu D, Lin H. Value iteration adaptive dynamic programming for optimal control of discrete-time nonlinear systems. *IEEE Trans Cybern* 2015;46:840-53. DOI
17. Luo B, Liu D, Huang T, Yang X, Ma H. Multi-step heuristic dynamic programming for optimal control of nonlinear discrete-time systems. *Inf Sci* 2017;411:66-83. DOI
18. Heydari A. Stability analysis of optimal adaptive control using value iteration with approximation errors. *IEEE Trans Automat Contr* 2018;63:3119-26. DOI
19. Li C, Ding J, Lewis FL, Chai T. A novel adaptive dynamic programming based on tracking error for nonlinear discrete-time systems. *Automatica* 2021;129:109687. DOI
20. Kiumarsi B, Alqaudi B, Modares H, Lewis FL, Levine DS. Optimal control using adaptive resonance theory and Q-learning. *Neurocomputing* 2019;361:119-25. DOI
21. Shang Y. Consensus tracking and containment in multiagent networks with state constraints. *IEEE Trans Syst Man Cybern Syst* 2023;53:1656-65. DOI
22. Shang Y. Scaled consensus and reference tracking in multiagent networks with constraints. *IEEE Trans Netw Sci Eng* 2022;9:1620-9. DOI
23. Zhang J, Yang D, Zhang H, Wang Y, Zhou B. Dynamic event-based tracking control of boiler turbine systems with guaranteed performance. *IEEE Trans Automat Sci Eng* 2023. DOI
24. Gao W, Mynuddin M, Wunsch DC, Jiang ZP. Reinforcement learning-based cooperative optimal output regulation via distributed adaptive internal model. *IEEE Trans Neural Netw Learn Syst* 2022;33:5229-40. DOI
25. Luo B, Liu D, Huang T, Liu J. Output tracking control based on adaptive dynamic programming with multistep policy evaluation. *IEEE Trans Syst Man Cybern Syst* 2017;49:2155-65. DOI
26. Wang D, Li X, Zhao M, Qiao J. Adaptive critic control design with knowledge transfer for wastewater treatment applications. *IEEE Trans Ind Inf* 2023. DOI
27. Ming Z, Zhang H, Li W, Luo Y. Neurodynamic programming and tracking control for nonlinear stochastic systems by PI algorithm. *IEEE Trans Circuits Syst II Express Briefs* 2022;69:2892-6. DOI
28. Lu J, Wei Q, Liu Y, Zhou T, Wang FY. Event-triggered optimal parallel tracking control for discrete-time nonlinear systems. *IEEE Trans Syst Man Cybern Syst* 2022;52:3772-84. DOI
29. Zhao M, Wang D, Ha M, Qiao J. Evolving and incremental value iteration schemes for nonlinear discrete-time zero-sum games. *IEEE Trans Cybern* 2023;53:4487-99. DOI
30. Al-Dabooni S, Wunsch DC. Online model-free *n*-step HDP with stability analysis. *IEEE Trans Neural Netw Learn Syst* 2020;31:1255-69. DOI
31. Ha M, Wang D, Liu D. A novel value iteration scheme with adjustable convergence rate. *IEEE Trans Neural Netw Learn Syst* 2023;34:7430-42. DOI
32. Luo B, Wu HN, Huang T. Optimal output regulation for model-free quanser helicopter with multistep Q-learning. *IEEE Trans Ind Electron* 2017;65:4953-61. DOI