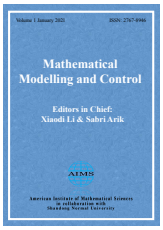
Mathematical Modelling and Control

https://www.aimspress.com/journal/mmc



MMC, 6(1): 97–110.
DOI: 10.3934/mmc.2026008
Received: 17 April 2024
Revised: 17 June 2024
Accepted: 26 August 2024
Published: 05 March 2026

Research article

Model-free optimal consensus control for multi-agent systems via DDPG-based event-triggered adaptive dynamic programming method

Pengfei Zhu[1], **Xiaolin Wang**[1,*], **Fangfei Li**[1,2,*], **Siyu Qian**[1] **and Haitao Li**[3]

[1] School of Mathematics, East China University of Science and Technology, Shanghai, China

[2] Key Laboratory of Smart Manufacturing in Energy Chemical Process, Ministry of Education, East China University of Science and Technology, Shanghai, China

[3] School of Mathematics and Statistics, Shandong Normal University, Jinan 250014, China

* **Correspondence:** Email: xiaolinwang@ecust.edu.cn, lifangfei@ecust.edu.cn, li_fangfei@163.com.

Abstract: This paper primarily addresses the design of distributed optimal cooperative controllers and the utilization of a reinforcement learning (RL)-based event-triggered mechanism for multi-agent systems (MASs) with unknown dynamics. By setting an extra compensator, the augmented system is constructed to overcome the dependence for system dynamics. Then, to address the issue of computational burden, we utilize an event-triggered mechanism based on reinforcement learning (RL) and neural networks (NNs) to implement the adaptive dynamic programming (ADP) algorithm. Additionally, we take into consideration the trade-off between computational burden and achieving consensus control by introducing a weighting factor in the reward design for MASs. With this reward design, we present an algorithm based on the deep deterministic policy gradient (DDPG) algorithm to learn the event-triggered condition for MASs and achieve a balance between these two factors. The event-triggered mechanism of our algorithm can also identify constraints such as time limitations or computational resource restrictions, aiming to achieve consensus control without violating these constraints. We demonstrate the absence of Zeno behavior and the uniform ultimate boundedness (UUB) of both local consensus error and weight estimation error. Finally, simulation results illustrate the effectiveness of the control algorithm and the weighting factor.

Keywords: augmented system; event-triggered adaptive dynamic programming; deep deterministic policy gradient; event-triggered condition

1. Introduction

In recent years, distributed control of multi-agent systems (MASs) has drawn widespread concern in different fields, such as traffic signal control [1], wheeled mobile robotic systems [2], smart grids [3], formation control [4], etc. The optimal consensus problem, as a fundamental control issue in MASs, aims to achieve consensus among different agents while minimizing computational performance consumption. It is widely acknowledged that solving the optimal consensus problem involves addressing the coupled Hamilton-Jacobi-Bellman (HJB) equation, which is difficult to solve analytically. Adaptive dynamic programming (ADP), as a powerful tool for optimizing control of complex systems, is proposed to tackle the problem.

ADP shares a core idea with reinforcement learning (RL)-finding optimal control strategies to minimize value functions. Recently, the research of ADP and its related fields has attracted extensive attention. Such as in [5], an ADP algorithm is proposed to address the data-driven zero-sum neuro-optimal control problem in continuous-time systems with unknown dynamics. In [6], the iterative ADP algorithm is employed to tackle infinite horizon undiscounted optimal control problems and to study discrete-time HJB equations. [7] proposes an integral

reinforcement learning algorithm based on ADP to obtain the iterative control with unknown disturbances. [8] designs an adaptive fuzzy controller for multivariable nonlinear systems with uncertainty. In [9], an ADP algorithm is utilized to solve the fault-tolerant control problem of a hydraulic servo actuator in the presence of actuator faults.

However, in practical scenarios, the unknown or partially known underlying dynamics of a system pose a challenge for ADP. Accurate estimation of value functions and optimal control decisions becomes challenging in the absence of precise knowledge about system dynamics. Various techniques, such as augmented system and online learning, have been developed to tackle the challenge of unknown dynamics and enhance the adaptability of ADP algorithms in real-world applications. In [10, 11], the augmented system is employed to overcome the requirement for explicit knowledge of system dynamics. [12] presents an RL method that tackles the problem of optimal stabilization in the presence of unknown dynamics. A recurrent neural network (NN) is introduced to reconstruct the nonlinear system in [13].

Moreover, ADP algorithms typically require extensive calculations and iterations, resulting in a substantial computational burden. This limitation hinders their practical usage, particularly in real-time control systems where efficiency is paramount. In order to alleviate the high computational cost, a new data sampling framework, namely the event-triggered control mechanism, is designed out. The fundamental idea of this approach is that controllers are triggered based on specific events or conditions, rather than executing control actions continuously. Event-triggered adaptive dynamic programming (ETADP), is studied in [14] with the combination of ADP and the event-triggered control mechanism. It can be widely applied in continuous-time systems [15–17] and discrete-time systems [18–20]. Among numerous research methodologies, the employment of RL in studying event-triggered control has gradually become a research hotspot.

RL is a subfield of machine learning that focuses on selecting suitable actions to maximize the reward signal. RL has demonstrated promising results in numerous fields, including addressing optimal consensus problems in MASs. In [21–24], an adaptive RL method is designed to optimize the control performance for nonlinear MASs. RL is specifically designed to overcome the challenges of solving the HJB equation for second-order unknown non-linear dynamical MASs in [25]. Additionally, [26] proposes a model-free algorithm that utilizes RL to determine the feedback gain matrix. RL has also demonstrated remarkable effectiveness in the domain of event-triggered control mechanisms. For example, in [27], a Q-learning-based event-triggered mechanism is designed to deal with intermittent-DoS attacks and power loads. Compared to traditional event-triggered mechanisms, the RL-based event-triggered mechanisms offer a superior level of adaptability in tackling more intricate challenges. This is attributed to their ability to fine-tune the reward mechanism, facilitating adaptability across different environments. [28] proposes a specific form of the event-triggered thresholds to satisfy the uniform ultimate boundedness (UUB) of the consensus error, but focused solely on computational burden without considering the rate of achieving consensus control. How to strike a balance between computational burden and convergence speed in addressing the consensus problem of MASs is evidently important, but currently, no one has conducted research on this. Furthermore, in real-world applications, certain constraints may exist, such as limited computational resources or time limitations. Exploring methods to identify and prevent exceeding these constraints is also a valuable research area.

Motivated by the above analysis, this paper presents a novel approach for distributed optimal consensus control in MASs with unknown dynamics. An augmented system is utilized to overcome the dependence of the system dynamics. Due to the fact that the analytical solutions to the continuous-time HJB equations are difficult to calculate, an NN is used to approximate the value function and develop the control policies. Additionally, an algorithm based on the deep deterministic policy gradient (DDPG) method is designed to update the event-triggered condition. The main contributions of the paper are summarized as follows:

1) We present a DDPG-based ETADP method to solve the optimal consensus problem for MASs. Utilizing the augmented system eliminates the need to acquire system dynamics. Expanding upon prior studies on event-triggered mechanisms, we propose incorporating

a weighting factor into the reward design of Markov decision processes (MDPs) for MASs to balance the computational burden and the rate of achieving consensus control.

2) We present an algorithm based on the DDPG algorithm to learn the event-triggered condition for MASs. Compared to the traditional ETADP method, our approach has the ability to recognize and tackle complex scenarios, with the goal of achieving consensus control while adhering to specified constraints like computational resource limitations.
3) We provide the stability of the MASs and the avoidance of Zeno behavior. Under the DDPG-based event-triggered control mechanism, we demonstrate that the event-triggered interval has a positive lower bound and both the local consensus error and weight estimation error are UUB.

The rest of this paper is organized as follows. In Section 2, some graph theory knowledge and necessary notations are provided, and the problem formulation for MASs is derived. In Section 3, we develop a model-free algorithm based on the event-triggered mechanism and introduce NNs to implement this algorithm. In Section 4, we design the event-triggered mechanism through the DDPG method. In Section 5, a simulation example is introduced to display the feasibility of our method. Finally, we give a conclusion in Section 6.

Notations: In this paper, $\mathbb{R}^n$ stands for the n-dimensional vector space, and $\mathbb{R}^{n\times m}$ represents the space composed of the $n \times m$ dimensional matrix. $\top$ represents the transpose symbol. The Euclidean norm of the n-dimensional vector x is defined as $\|x\| = \sqrt{x^\top x}$. $\lambda_{min}(A)$ stands for the minimum eigenvalue of matrix A. $A > 0$ represents that A is a positive definite matrix, and $A \geq 0$ means that it's a semi-positive definite matrix. $\mathbb{E}[\cdot]$ represents the mathematical expectation.

2. Preliminaries

In this section, some necessary preliminaries and the problem formulation are introduced.

2.1. Graph theory

$\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathcal{A})$ is a directed communication graph, in which $\mathcal{V} = \{v_1, v_2, ..., v_N\}$ denotes the node set and $\mathcal{A} = [a_{ij}] \in \mathbb{R}^{n\times n}$ indicates the weighted adjacency matrix of $\mathcal{G}$. $\mathcal{E} = \left\{e_{ij} = \left(v_i, v_j\right), i \neq j\right\} \subseteq \mathcal{V} \times \mathcal{V}$ represents the edge set, and an edge $e_{ji} = (v_j, v_i) \in \mathcal{E}$ if and only if $a_{ij} > 0$, which means that v_j can send information to v_i. The neighbors of node v_i can be represented as $\mathcal{N}_i = \{v_j | (v_j, v_i) \in \mathcal{E}, v_i, v_j \in \mathcal{V}\}$. $\mathcal{D} = \text{diag}\{d_1, d_2, ..., d_N\}$ is the in-degree matrix, in which $d_i = \sum_{j\in\mathcal{N}_i} a_{ij}$ denotes the sum of edge weights originating from node i. The digraph Laplacian matrix $\mathcal{L}$ can then be obtained by subtracting the adjacency matrix $\mathcal{A}$ from the in-degree matrix $\mathcal{D}$, resulting in $\mathcal{L} = \mathcal{D} - \mathcal{A}$.

2.2. Consensus of MASs

We consider a linear MAS consisting of N identical agents, and the dynamics of each agent are

$$\dot{x}_i(t) = Ax_i(t) + B_i u_i(t), i = 1, 2, ..., N, \tag{2.1}$$

where $x_i(t) \in \mathbb{R}^n$ is the state of the agent i, $u_i(t) \in \mathbb{R}^m$ indicates the vector of control input, and both $A \in \mathbb{R}^{n\times n}$ and $B_i \in \mathbb{R}^{n\times m}$ are considered to be unknown.

A reference system, known as a leader, is defined as $x_0(t) \in \mathbb{R}^n$. In general, its trajectory satisfies the dynamics

$$\dot{x}_0(t) = Ax_0(t). \tag{2.2}$$

Taking into account the information interaction within the MAS, we introduce the concept of local neighborhood tracking errors, denoted as

$$\mathrm{e}_i(t) = \sum_{j\in\mathcal{N}_i} a_{ij}(x_i(t) - x_j(t)) + b_i(x_i(t) - x_0(t)), \tag{2.3}$$

where the pinning gain, denoted as $b_i \geq 0$, represents the extent of connection. Specifically, $b_i > 0$ is satisfied if and only if there is a directed path from the leader to the ith agent, indicating their interaction. Conversely, when such a path does not exist, we set $b_i = 0$, indicating the absence of interaction.

Remark 2.1. *Marked for review in case this referenced equation and others in the paper need to be linked. According to the above expression, the consensus*

information of MAS (2.1) can be represented by the local neighborhood consensus error e_i. We say that agent i achieves consensus with the leader if $\lim_{k\to\infty} \|e_i(t)\| = 0$.

Considering Eq (2.3), the global tracking error vector can be expressed as

$$\begin{aligned} e(t) &= (\mathcal{L} \otimes I_n)\, x(t) + (\mathcal{B} \otimes I_n)\,(x(t) - \bar{x}_0(t)) \\ &= (\mathcal{L} \otimes I_n)\, x(t) - (\mathcal{L} \otimes I_n)\, \bar{x}_0(t) + (\mathcal{B} \otimes I_n)\,(x(t) - \bar{x}_0(t)) \\ &= ((\mathcal{L} + \mathcal{B}) \otimes I_n)\,(x(t) - \bar{x}_0(t)), \end{aligned} \tag{2.4}$$

where $\mathcal{B} = \mathrm{diag}\{b_1, b_2, ..., b_N\}$, $\mathcal{L}$ refers to the digraph Laplacian matrix, $\bar{x}_0(t) = [x_0^\top(t), x_0^\top(t), ..., x_0^\top(t)]^\top$, $e(t) = \left[e_1^\top(t), e_2^\top(t), ..., e_N^\top(t)\right]^\top$, $x(t) = \left[x_1^\top(t), x_2^\top(t), ..., x_N^\top(t)\right]^\top$, and I_n represents the identity matrix.

After differentiating (2.3), the dynamics of the local neighborhood tracking error can be represented as

$$\begin{aligned} \dot{e}_i(t) &= \sum_{j=1}^{N} \left(l_{ij} + b_{ij}\right)\left(A\left(x_j - x_0\right) + B_j u_j\right) \\ &= (l_{ii} + b_{ii})\,(A\,(x_i - x_0) + B_i u_i) \\ &\quad + \sum_{j\in\mathcal{N}_i} \left(l_{ij} + b_{ij}\right)\left(A\left(x_j - x_0\right) + B_j u_j\right), \end{aligned} \tag{2.5}$$

where $b_{ij} = \begin{cases} b_i, & i = j. \\ 0, & i \neq j. \end{cases}$

Thus, we can condense the consensus problem of (2.5) to crafting a viable policy that can satisfy $\lim_{k\to\infty} \|e_i(t)\| = 0$.

2.3. Problem formulation

In this paper, we will design a method to achieve consensus control while solving the following problems:

1) The system dynamics in (2.5) are unknown and our method needs to overcome the dependence on system dynamics.
2) The event-triggered mechanism of the method requires sufficient flexibility and strives to operate within the specified constraints as much as possible. It should possess the ability to strike a balance between the computational burden and the rate of achieving consensus control. Moreover, it can adapt to complex scenarios, including achieving consensus control within time constraints or with limited computational resources. In this work, we indirectly assess the consumption of computational resources by tracking the frequency of control updates carried out by each agent.
3) This method requires ensuring the stability of system dynamics as well as its event-triggered mechanism to prevent the occurrence of Zeno behavior phenomena.

3. Model-free optimal control design

A method is introduced to describe the MAS with unknown system dynamics in this section. Meanwhile, a model-free optimal control is applied to minimize the performance cost. Furthermore, NNs and an event-triggered mechanism are designed to implement the ADP scheme.

3.1. Augmented system

In order to overcome the requirements of system dynamics, a controllable system that combines the tracking error system (2.5) and the pre-compensator is designed. The pre-compensator is a component that modifies the input signal to enhance the overall performance of the system. First, the compensator is defined as

$$\dot{u}_i = a_i\,(u_i) + b_i\,(u_i)\,\omega_i, \tag{3.1}$$

where $u_i \in \mathbb{R}^m$ represents the input vector of the ith agent. a_i and b_i are two functions designed to fulfill the requirements of a controllable system.

Combining (2.5) and (3.1), the tracking error system can be represented as

$$\dot{\bar{e}}_i = F_i\,(\bar{e}_i) + G_i\,(\bar{e}_i)\,\omega_i, \tag{3.2}$$

where $\bar{e}_i = \begin{bmatrix} e_i \\ u_i \end{bmatrix} \in \mathbb{R}^{m+n}$ represents the state vector, $\omega_i \in \mathbb{R}^m$ serves as the control input of the augmented system, and the partitioned matrix F_i and G_i can be written as

$$F_i(\bar{e}_i) = \begin{bmatrix} \sum\limits_{j=1}^{N} \left(l_{ij} + b_{ij}\right)\left(A\left(x_j - x_0\right) + B_j u_j\right) \\ a_i(u_i) \end{bmatrix} \in \mathbb{R}^{m+n}$$

and

$$G_i(\bar{e}_i) = \begin{bmatrix} 0 \\ b_i(u_i) \end{bmatrix} \in \mathbb{R}^{(m+n)\times m},$$

where $\|G_i(\bar{e}_i)\| \leq G_{i,max}$.

Define a continuous performance index, also known as a cost function, which needs to be optimized as follows:

$$J_i(\bar{e}_i, \omega_i) = \int_0^\infty (\bar{e}_i^\top Q_i \bar{e}_i + \omega_i^\top R_{ii} \omega_i) d\tau, \tag{3.3}$$

where $Q_i > 0$, $R_{ii} > 0$.

Thus, we convert the task of designing the control u_i into the problem of designing the consensus control ω_i.

3.2. ADP algorithm on MASs

Definition 3.1 (admissible control)**.** *The control $\omega_i \in \Omega_i$ is defined as admissible if the following holds:*

1) ω_i keeps continuous on Ω_i and $\omega_i(0) = 0$.
2) ω_i ensures the stability of the MAS.
3) $\int_{t_0}^\infty (\bar{e}_i^\top Q_i \bar{e}_i + \omega_i^\top R_{ii} \omega_i) d\tau < \infty$.

On the basis that ω_i is an admissible control, the local value function of the agent i can be defined as follows:

$$V_i(\bar{e}_i) = \int_t^\infty (\bar{e}_i^\top Q_i \bar{e}_i + \omega_i^\top R_{ii} \omega_i) d\tau. \tag{3.4}$$

Then, we can establish the HJB equation

$$\begin{aligned} H_i(\bar{e}_i, \nabla V_i, \omega_i) &= \bar{e}_i^\top Q_i \bar{e}_i + \omega_i^\top R_{ii} \omega_i + \dot{V}_i \\ &= \bar{e}_i^\top Q_i \bar{e}_i + \omega_i^\top R_{ii} \omega_i + \nabla V_i^\top (F_i(\bar{e}_i) + G_i(\bar{e}_i)\omega_i) \\ &= 0, \end{aligned} \tag{3.5}$$

where ∇V_i is the gradient of the local value function $V_i(\bar{e}_i)$. By using the first-order necessary condition $\partial H_i / \partial \bar{e}_i = 0$, we obtain the optimal control policies as

$$\omega_i^* = -\frac{1}{2} R_{ii}^{-1} G_i^\top(\bar{e}_i) \nabla V_i^*, \tag{3.6}$$

where V_i^* represents the optimal value function and $\nabla V_i^* = \partial V_i^* / \partial \bar{e}_i$. When using the optimal control policies (3.6) for agent i, the HJB equation can be expressed as

$$\begin{aligned} H_i(\bar{e}_i, \nabla V_i^*, \omega_i^*) = &-\frac{1}{4} (\nabla V_i^*)^\top G_i(\bar{e}_i) R_{ii}^{-1} G_i^\top(\bar{e}_i) \nabla V_i^* \\ &+ \bar{e}_i^\top Q_i \bar{e}_i + (\nabla V_i^*)^\top F_i(\bar{e}_i) \\ =&0. \end{aligned} \tag{3.7}$$

Through the aforementioned expressions, we can acquire the optimal control (3.6) by solving the HJB equation (3.7). However, solving the HJB equation analytically is a difficult task due to its nonlinear and high-dimensional characteristics. Furthermore, updating the controller continuously would lead to a depletion of computational resources. All of these problems contribute to the difficulty in achieving consensus control while reducing computational burden.

3.3. ETADP method via NNs

In this subsection, we present the event-triggered mechanism for reducing computational load. The controller ω_i can solely be updated at specific moments called event-triggered sampling instants $\{t_i^k, k = 0, 1, 2, ...\}$, where t_i^k denotes the kth event-triggered sampling instant.

Therefore, the system (3.2) can be rewritten as

$$\dot{\bar{e}}_i = F_i(\bar{e}_i) + G_i(\bar{e}_i) \hat{\omega}_i, \tag{3.8}$$

where $\hat{\omega}_i$ represents the event-triggered controller for agent i.

Assumption 3.1. *We assume that the partitioned matrix F_i satisfies the Lipschitz condition when $t \in [t_i^k, t_i^{k+1})$, i.e., $\left\|F_i(\bar{e}_i(t_i^k)) - F_i(\bar{e}_i(t))\right\| \leq L_{F_i} \left\|\bar{e}_i(t_i^k) - \bar{e}_i(t)\right\|$, where L_{F_i} denotes the Lipschitz constant associated with the partitioned matrix F_i.*

Considering that the analytical solution of the HJB equations is difficult to obtain, we employ the following NN to approximate the value function:

$$V_i(\bar{e}_i) = W_i^\top \Phi_i(\bar{e}_i) + \varepsilon_i, \tag{3.9}$$

where W_i serves as the weight vector, Φ_i represents the activation function, and ε_i denotes the reconstruction error. We define the residual error as

$$\sigma_i = \bar{e}_i^\top Q_i \bar{e}_i + \omega_i^\top R_{ii} \omega_i + W_i^\top \nabla \Phi_i \dot{\bar{e}}_i, \tag{3.10}$$

where $\nabla \Phi_i = \frac{\partial \Phi_i}{\partial \bar{e}_i^\top}$. For agent i, the estimated value function can be represented as

$$\hat{V}_i(\bar{e}_i) = \hat{W}_i^\top \Phi_i(\bar{e}_i), \tag{3.11}$$

where $\hat{W}_i$ stands for the weight estimation vector. We define the estimated residual error as

$$\hat{\sigma}_i = \bar{e}_i^\top Q_i \bar{e}_i + \hat{\omega}_i^\top R_{ii} \hat{\omega}_i + \hat{W}_i^\top \nabla \Phi_i \dot{\bar{e}}_i. \tag{3.12}$$

The error performance index can be defined as

$$E_i = \frac{1}{2}\hat{\sigma}_i^\top \hat{\sigma}_i. \tag{3.13}$$

In order to minimize the error performance index E_i, we employ the gradient descent method to update the weight estimation error. When $t = t_i^k$,

$$\begin{aligned}\dot{\hat{W}}_i &= -\alpha_i \frac{\partial E_i}{\partial \hat{\sigma}_i}\frac{\partial \hat{\sigma}_i}{\partial \hat{W}_i} \\ &= -\alpha_i \nabla\Phi_i \dot{\bar{e}}_i \left(\bar{e}_i^\top Q_i \bar{e}_i + \hat{\omega}_i^\top R_{ii}\hat{\omega}_i + (\nabla\Phi_i \dot{e}_i)^\top \hat{W}_i\right)\end{aligned} \tag{3.14}$$

and

$$\dot{\hat{W}}_i = 0, t \in (t_i^k, t_i^{k+1}), \tag{3.15}$$

where α_i represents the learning rate. The weight estimation error, denoted as $\bar{W}_i = W_i - \hat{W}_i$, represents the deviation between the estimated weight and the true weight of the value function. Combining (3.6) and (3.11), the controller can be represented as

$$\hat{\omega}_i = -\tfrac{1}{2}R_{ii}^{-1}G_i^\top \nabla\Phi_i^\top \hat{W}_i. \tag{3.16}$$

Remark 3.1. *Since the system dynamics in MASs are unknown, and we need to overcome this dependence, we utilize an augmented system to eliminate the necessity of acquiring the system dynamics. Based on the provided statement, we can update the controller $\hat{\omega}_i$ through the weight estimation errors $\hat{W}_i$ and the partitioned matrix G_i. Consequently, the controller eliminates the dependence on A and B_i, thereby solving problem 1.*

To facilitate the subsequent proof of the theorem, we make the following assumptions.

Assumption 3.2. *Within the time interval when the event trigger occurs, controller ω_i satisfies the Lipschitz condition when $t \in [t_i^k, t_i^{k+1})$, i.e., $\|\omega_i - \hat{\omega}_i\| \le L_{\omega_i}\left\|\bar{e}_i - \bar{e}_i(t_i^k)\right\|$, where the symbol L_{ω_i} represents the Lipschitz constant related to the controller ω_i.*

Assumption 3.3. *We make the assumption that the weight matrix, the residual error, the gradient of the activation function, the reconstruction error and the gradient of the reconstruction error, $\nabla\Phi_i, \dot{\bar{e}}_i$ are bounded, that is, $\|W_i\| \le W_{i,max}$, $\|\sigma_i\| \le \sigma_{i,max}$, $\|\nabla\Phi_i\| \le \nabla\Phi_{i,max}$, $\|\varepsilon_i\| \le \varepsilon_{i,max}$, $\|\nabla\varepsilon_i\| \le \nabla\varepsilon_{i,max}$, and $\varphi_{i,min} \le \left\|\nabla\Phi_i \dot{\bar{e}}_i\right\| \le \varphi_{i,max}$.*

4. The DDPG-based event-triggered mechanism design

In this section, we present a threshold design approach that leverages the DDPG algorithm. Our proposed method aims to achieve optimal control while simultaneously minimizing the computational burden involved.

4.1. Design of the MDP

In the realm of deep reinforcement learning (DRL), the MDP serves as a fundamental framework for capturing the interaction dynamics between an agent and its environment. The state s_t, action a_t and reward r_t are defined as follows:

1) To incorporate both the computational burden and the rate of achieving consensus control, the state, denoted as s_t, should encompass information on these aspects. To achieve this objective, the consensus error of agents is quantified by recording it as

$$\bar{e}(t) = [\bar{e}_1^\top(t), \bar{e}_2^\top(t), ..., \bar{e}_N^\top(t)]^\top, \tag{4.1}$$

while the consensus error at the last event-triggered instant is denoted as

$$\bar{e}^k(t) = [(\bar{e}_1(t_1^k))^\top, (\bar{e}_2(t_2^k))^\top, ..., (\bar{e}_N(t_N^k))^\top]^\top. \tag{4.2}$$

Therefore, we design the state of the MDP as

$$s_t = [\bar{e}^\top(t), (\bar{e}^k(t))^\top]^\top. \tag{4.3}$$

2) The action a_t is determined by the DDPG algorithm in our research. Specifically, we define a_t as the threshold of the event-triggered mechanism. To ensure a bounded action space, we set an upper limit for each dimension of a_t, denoted as $a_{i,max}$. When evaluating the event-triggered condition for agent i, if $\left\|\bar{e}_i(t) - \bar{e}_i(t_i^k)\right\| \ge a_t(i)$, it indicates a violation of the event-triggered condition.
3) Our approach to reward allocation involves minimizing computational burden while ensuring consensus control. Compared to traditional methods, Algorithm 1 takes certain constraints or goals into account and reflects them in the reward, enabling it to adapt to more complex scenarios. First, we design an indicator

$$flag(t) = [flag_1(t), flag_2(t), ..., flag_N(t)]^\top \tag{4.4}$$

to reveal whether each agent violates the event-triggered condition at the current moment, in which

$$flag_i(t) = \begin{cases} -1, & \left\|\bar{e}_i(t) - \bar{e}_i(t_i^k)\right\| \geq a_t(i). \\ 0, & \text{else.} \end{cases} \tag{4.5}$$

Thus, we set the daily reward r_t^s and the reward at the termination of an episode r^f as

$$r_t^s = \frac{\frac{2}{\pi}\sum\limits_{i=1}^{N}\lambda_i \arctan(\frac{\|\bar{e}_i^*\|}{\|\bar{e}_i\|}) + \sum\limits_{i=1}^{N}(1-\lambda_i)flag_i}{N}, \tag{4.6}$$

$$r^f = \begin{cases} -C_1, & \text{if constraint conditions are violated,} \\ C_2, & \text{else,} \end{cases} \tag{4.7}$$

where $\left\|\bar{e}_i^*\right\|$ is a constant representing the signal indicating whether consensus control is satisfied and C_1 and C_2 are two nonnegative constants.

We consider consensus control in MASs to be achieved only when the $\|\bar{e}_i\|$ for each agent remains below $\left\|\bar{e}_i^*\right\|$ throughout the remaining duration. To mitigate the issue of the ratio $\frac{\|\bar{e}_i^*\|}{\|\bar{e}_i\|}$ approaching infinity in the future, we address this problem by incorporating the arctan function.

To balance the computational burden and the rate of achieving consensus control, each agent i is assigned a constant weight coefficient $\lambda_i \in (0, 1)$. The value of λ_i determines the impact they have on the long-term reward. If a lower computational burden is desired, the value of λ_i should be decreased. On the other hand, increasing the value of λ_i would prioritize achieving consensus control over reducing computational burden. Properly adjusting the value of λ_i allows for flexibility in balancing the trade-off between computational efficiency and consensus achievement. To enhance the influence of the constraints in reward settings, we set C_1 and C_2 to be relatively large values.

Fianlly, we have the reward r_t as

$$r_t = r_t^s + is^{final} r^f, \tag{4.8}$$

where is^{final} is a boolean value that is 1 only at the termination of an episode and 0 otherwise.

Remark 4.1. *According to (4.6), the MDP in Algorithm 1 takes into account both the rate of achieving consensus control and the computational burden, while (4.7) aids the system (3.8) in identifying the limiting conditions. Thus, we can solve problem 2 through the design of the MDP.*

Algorithm 1: DDPG-based ETADP method.

Initialize the weight $\hat{W}_i^0$, the critic network $Q(s, a|\theta^Q)$, the actor $\mu(s|\theta^\mu)$ and the target networks Q' and μ'
for *episode = 1:n* **do**
  Initialize a random process $\mathcal{N}$, and the original observation state s_1
  Initialize the number of event-triggering occurrences *num*
  for *t = 1:T* **do**
    Choose action $a_t = \mu(s_t|\theta^\mu) + \mathcal{N}_t$ according to the policy and noise;
    calculate the reward r_t and observe the next state s_{t+1};
    Store (s_t, a_t, r_t, s_{t+1}) in the replay buffer;
    Choose a random minibatch $B = (s_j, a_j, r_j, s_{j+1})$ from the replay buffer;
    Update the critic network through gradient ascent by (4.10);
    Update the actor policy through policy gradient by (4.11);
    Update the weights of the networks by (4.12);
    for *i = 1:N* **do**
      if $\left\|\bar{e}_i - \bar{e}_i^k\right\| < a_t(i)$ **then**
        Let $\hat{\omega}_i = \hat{\omega}_i^k$ and $\hat{W}_i = \hat{W}_i^k$
      else
        Update the weight $\hat{W}_i$ through

$$\hat{W}_i^{k+1} = \hat{W}_i^k - \alpha_i \nabla\Phi_i \dot{\bar{e}}_i^k \times \left((\bar{e}_i^k)^\top Q_i \bar{e}_i^k + (\hat{\omega}_i^k)^\top R_{ii} \hat{\omega}_i^k + (\nabla\Phi_i \dot{\bar{e}}_i)^\top \hat{W}_i^k\right) \tag{4.9}$$

        Update the controller $\hat{\omega}_i$ by (3.16);
        $num(i) = num(i) + 1$;
  Until convergence.

4.2. Deep deterministic policy gradient algorithm

To ensure continuity in the action space, we use the DDPG algorithm instead of the deep Q-learning (DQN)

algorithm. By incorporating both the rate of achieving consensus control and the computational burden into the Q-value function, we can effectively balance these two factors during the learning process. This enables the DDPG algorithm to make informed decisions that optimize both consensus achievement and computational burden. The Q-value function can be obtained as follows

$$Q(s_t, a_t) = \mathbb{E}\left[\sum_{i=t}^{T} \gamma^{i-t} r(s_t, a_t)\right], \tag{4.10}$$

where $\gamma \in (0, 1]$. In this paper, we prefer to set a relatively large value for γ in considerion of the long time steps. This means that instead of solely focusing on immediate rewards or short-term gains, the algorithm is more concerned about the potential consequences of actions in the future.

DDPG is developed using the actor-critic method, where the critic network approximates the Q-function and the action network approximates the action. In addition, DDPG employs separate target networks to compute the loss function. We define the critic network loss function L as

$$L\left(\theta^{Q}\right) = \frac{1}{N}\sum_{i}\left(r_i + \gamma Q'(s_{i+1}, \mu') - Q(s_i, a_i|\theta^{Q})\right)^2, \tag{4.11}$$

where Q' and Q are the target critic network and the critic network respectively, and θ^{Q}, θ^{μ} and $\theta^{Q'}$ represent the parameters of critic network, actor network, and target critic network, respectively. The actor network is updated through the policy gradient:

$$\nabla\theta^{\mu} J = \frac{1}{N}\sum_{i} \nabla_a Q\left(s_i, a_i \mid \theta^{Q}\right) \nabla_{\theta^{\mu}} \mu\left(s_i \mid \theta^{\mu}\right). \tag{4.12}$$

Soft update is used to update the parameters of the critic network and the actor network:

$$\theta' = \varepsilon\theta + (1 - \varepsilon)\theta', \tag{4.13}$$

where ε is a constant, and $0 < \varepsilon \leq 1$.

After incorporating the DDPG algorithm into ETADP control, we propose Algorithm 1 and obtain the following theorems.

Theorem 4.1. *When Assumptions 1–3 are satisfied, if we use (19) to update the controllers and use Algorithm 1 to learn the event-triggered condition, the weight estimation error $\bar{W}_i$ and the consensus error $\bar{e}_i$ are UUB.*

Proof. Please see Appendix A. □

Theorem 4.2. *Let Assumptions 1–3 hold. Under the event-triggered scheme based on Algorithm 1, the Zeno behavior will not occur, i.e., the event-triggered interval has a positive lower bound.*

Proof. Please see Appendix B. □

Remark 4.2. *Through Theorems 1 and 2, we prove the stability of the MASs and the avoidance of Zeno behavior, thereby solving problem 3.*

5. Simulation results

In this section, one example is taken to display the effectiveness of our proposed method. Based on the structure depicted in Figure 1, we can observe that the MAS consists of a leader and three followers. We set the weighting matrices in the local cost functions as

$$Q_1 = Q_2 = Q_3 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix},$$

$$R_{ij} = \begin{cases} 2, & i = j, \\ 1, & i \neq j. \end{cases}$$

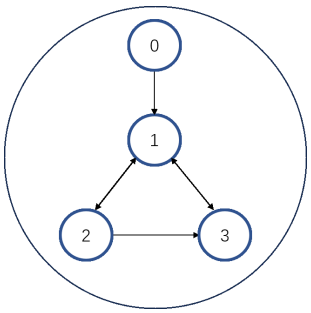


Figure 1. Structure of the multi-agent systems.

For agent i, let the activation functions be

$$\Phi_i(\bar{e}_i) = [\tanh(\bar{e}_i^2(1)), \tanh(\bar{e}_i(1)\bar{e}_i(2)), \tanh(\bar{e}_i^2(2))]^{\top}.$$

We use the variable $num \in \mathbb{R}^N$ to count the number of event-triggering occurrences, which serve as a metric for evaluating the computational burden. We impose restrictions on computational resources, specifically, $\sum_{i=1}^{3} num(i) \leq 300$.

Additionally, we select the variable *time* $\in \mathbb{R}^N$ to address the rate at which consensus control is achieved. We select $a_i(\tau_i) = -2\tau_i, b_i(\tau_i) = \cos^2(\tau_i), i = 1, 2, 3$. We set the reward parameter at the end of the episode as $C_1 = C_2 = 100$. In this experiment, we assign the same value, denoted as λ, to all the weighting factors. In the design of Algorithm 1, we choose $\lambda = 0.4$ and $\left\|\bar{e}_i^*\right\| = 0.01, i = 1, 2, 3$. Then we change the value of λ to 0.2 and 0.6, respectively.

The MAS models are designed as

$$\dot{x}_0 = Ax_0, \dot{x}_i = Ax_i + B_i u_i, i = 1, 2, 3,$$

where $A = \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix}$, $B_1 = \begin{bmatrix} 2 \\ 1 \end{bmatrix}$, $B_2 = \begin{bmatrix} 2 \\ 3 \end{bmatrix}$ and $B_3 = \begin{bmatrix} 2 \\ 2 \end{bmatrix}$.

In this experiment, we evaluate both the computational burden and the rate of achieving consensus control using the ETADP control algorithm in MASs. Figures 2 and 3 illustrate the evolution of agents' states under ETADP control, with the color of each line representing a distinct agent. We can observe that system state x_i closely follows the trajectories of the leader x_0 after about 15 seconds. This demonstrates the effectiveness of the algorithm in achieving consensus control. From Figure 4, we observe that the weight matrix W_i stabilizes after about 10 seconds. Upon observation, it is evident that utilizing Algorithm 1 to design an event-triggered mechanism still satisfies the UUB condition for W_i. Figure 5 shows the admissible control of all agents under the event-triggered mechanism.

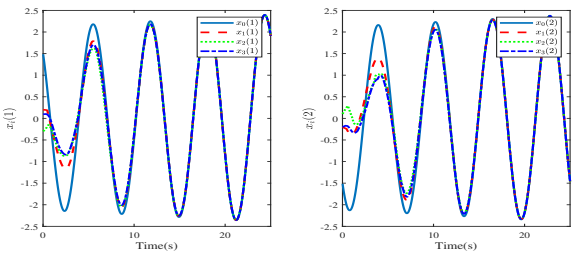


Figure 2. States of all agents.

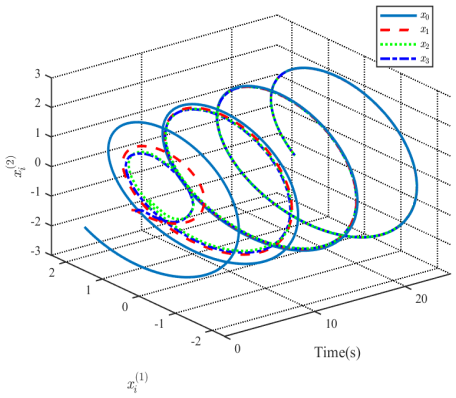


Figure 3. 3D phase plane plot.

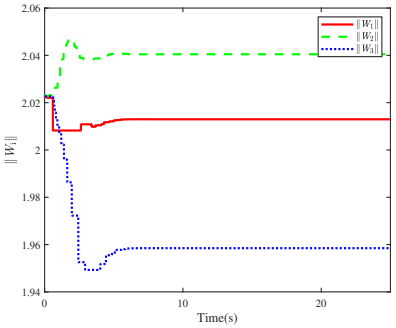


Figure 4. Weight matrices of all agents.

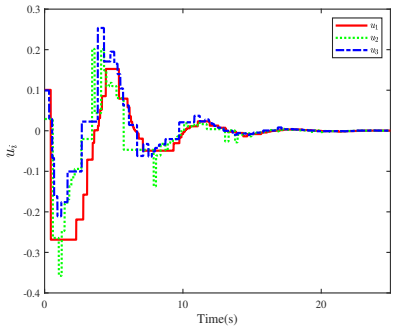
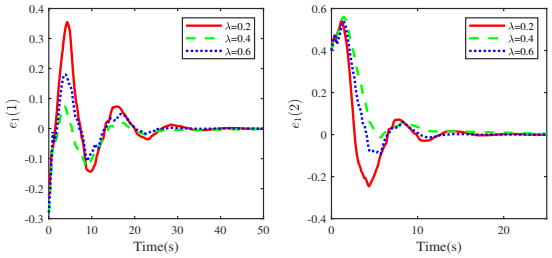


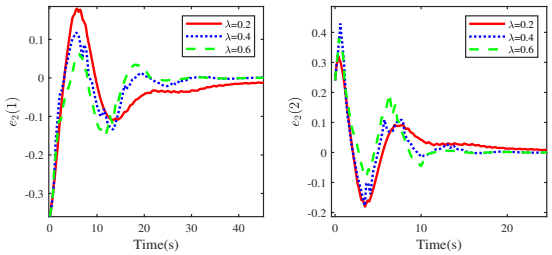
Figure 5. Controls of all agents.

Next, we evaluate the effect of weighting factors λ_i on both reducing the computational burden and achieving consensus control. Table 1 shows the computational burden associated with various λ_i. Table 2 displays the time required to achieve consensus control. Figure 6 depicts the evolution of consensus errors, with the color of each line representing a different λ_i. From Table 1, it is evident that there are no violations of the constraints at different values of λ. From

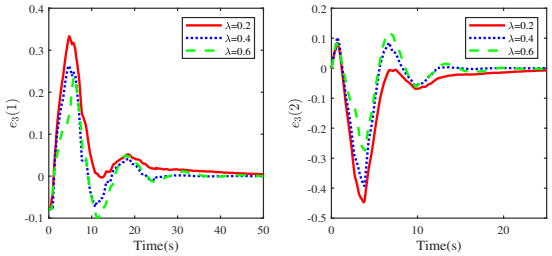
Figure 6 and Tables 1 and 2, it can be seen that the higher values of λ_i lead to earlier achievement of consensus control and more frequent event triggers.



(a) Consensus errors with different λ_i for agent 1



(b) Consensus errors with different λ_i for agent 2



(c) Consensus errors with different λ_i for agent 3

Figure 6. Consensus errors of all agents.

Table 1. Comparisons of different λ.

λ	$num(1)$	$num(2)$	$num(3)$	$\sum_{i=1}^{3} num(i)$
0.2	77	70	56	203
0.4	98	77	71	246
0.6	115	94	79	288

Table 2. The time required to achieve consensus control.

λ	$time(1)(s)$	$time(2)(s)$	$time(3)(s)$
0.2	30.94	37.02	36.31
0.4	20.12	28.11	22.45
0.6	18.73	25.69	21.82

6. Conclusions

In this study, we propose a novel DDPG-based approach to achieve optimal consensus control for MASs with unknown dynamics. An innovative aspect of our work lies in the utilization of the DDPG algorithm to design the event-triggered mechanism for MASs. When designing the reward function in Algorithm 1, we take into account both the computational burden and the rate of achieving consensus control. Additionally, we introduce a weight parameter, denoted as λ_i, which allows us to strike a balance between these two objectives. The event-triggered mechanism can also recognize and tackle complex situations, aiming to guarantee consensus control under certain restrictions. Under the event-triggered scheme based on Algorithm 1, we employ a Lyapunov function to prove the UUB condition of the weight estimation error $\bar{W}_i$ and the consensus error $\bar{e}_i$. Then we prove that the event-triggered interval occurs as a positive lower bound to ensure the elimination of the Zeno behavior for MASs. Finally, we perform simulations to verify the effectiveness of our proposed method in achieving optimal consensus control. The results of the simulations affirm the efficacy and feasibility of the approach.

Use of Generative-AI tools declaration

The authors declare they have not used Artificial Intelligence (AI) tools in the creation of this article.

Acknowledgments

This work is supported by the National Natural Science Foundation of China under Grants 62573198, 62173142, 62303185, National Key Laboratory of Space Target Awareness under Grant STA2025ZCB0208 and the

Shanghai Sailing Program under the grant 23YF1409500.

Conflict of interest

The authors declare that there are no conflicts of interest with other works in this paper.

Appendix

Appendix A. Proof of Theorem 1

To prove the theorem, we will divide it into two cases for discussion.

Case 1: When it does not exceed the threshold conditions learned by Algorithm 1, i.e., $t \in (t_i^k, t_i^{k+1})$. We define the Lyapunov function candidate L_i as

$$L_i = L_{i1} + L_{i2}, \tag{1}$$

where $L_{i1} = V^*(\bar{e}_i)$ and $L_{i2} = \frac{1}{2} tr(\bar{W}_i^\top \bar{W}_i)$. Taking the first derivative of L_i, one has

$$\dot{L}_i = \dot{V}^*(\bar{e}_i) + tr\left(\bar{W}_i^\top \dot{\bar{W}}_i\right). \tag{2}$$

Considering that $t \in (t_i^k, t_i^{k+1})$, we can obtain that $\dot{\bar{W}}_i = 0$. Therefore,

$$\begin{aligned} \dot{L}_i &= \dot{V}^*(\bar{e}_i) \\ &= (\nabla V_i^*)^\top (F_i(\bar{e}_i) + G_i(\bar{e}_i)\hat{\omega}_i) \\ &= (\nabla V_i^*)^\top (F_i(\bar{e}_i) + G_i(\bar{e}_i)\omega_i^*) \\ &\quad - (\nabla V_i^*)^\top G_i(\bar{e}_i)(\omega_i^* - \hat{\omega}_i). \end{aligned} \tag{3}$$

According to (3.5), we can obtain that

$$\begin{aligned} &(\nabla V_i^*)^\top \left(F_i(\bar{e}_i) + G_i(\bar{e}_i)\,\omega_i^*\right) \\ =& -\bar{e}_i^\top Q_i \bar{e}_i - \omega_i^{*\top} R_{ii} \omega_i^*. \end{aligned} \tag{4}$$

Therefore, combining (3) and (4), one has

$$\begin{aligned} \dot{L}_i = &-\bar{e}_i^\top Q_i \bar{e}_i - \omega_i^{*\top} R_{ii} \omega_i^* \\ &- (W_i^\top \nabla\Phi_i + \nabla\varepsilon_i) G_i(\bar{e}_i)(\omega_i^* - \hat{\omega}_i), \end{aligned} \tag{5}$$

where $\nabla\Phi_i = \frac{\partial \Phi_i}{\partial \bar{e}_i^\top}$ and $\nabla\varepsilon_i = \frac{\partial \varepsilon_i}{\partial \bar{e}_i^\top}$. Due to the fact that the controller ω_i satisfies the Lipschitz condition, i.e., $\|\omega_i - \hat{\omega}_i\| \le L_{\omega_i} \left\|\bar{e}_i - \bar{e}_i(t_i^k)\right\|$, one has

$$\begin{aligned} \dot{L}_i &\le -\lambda_{min}(Q_i)\|\bar{e}_i\|^2 + \frac{1}{2}\left\|W_i^\top \nabla\Phi_i + \nabla\varepsilon_i\right\|^2 \\ &\quad + \frac{1}{2}\left\|G_i(\bar{e}_i)(\omega_i^* - \hat{\omega}_i)\right\|^2 \\ &\le -\lambda_{min}(Q_i)\|\bar{e}_i\|^2 + \frac{1}{2}\left\|W_i^\top \nabla\Phi_i + \nabla\varepsilon_i\right\|^2 \\ &\quad + \frac{1}{2} G_{i,max}^2 L_{\omega_i}^2 \left\|\bar{e}_i - \bar{e}_i(t_i^k)\right\|^2. \end{aligned} \tag{6}$$

Since it does not exceed the threshold conditions, we have

$$\left\|\bar{e}_i(t_i) - \bar{e}_i(t_i^k)\right\| < a_t(i), \tag{7}$$

where $a_t(i)$ represents the threshold for agent i. When designing the action of the Algorithm 1, we set $a_t(i)$ to be bounded, thus we have

$$\begin{aligned} \dot{L}_i \le &-\lambda_{min}(Q_i)\|\bar{e}_i\|^2 + W_{i.max}^2 \nabla\Phi_{i,max}^2 \\ &+ \nabla\varepsilon_{i,max}^2 + \frac{1}{2} G_{i,max}^2 L_{\omega_i}^2 a_{i,max}^2. \end{aligned} \tag{8}$$

To guarantee $\dot{L}_i < 0$, the following conditions should hold:

$$\|\bar{e}_i\| > \sqrt{\frac{W_{i.max}^2 \nabla\Phi_{i,max}^2 + \nabla\varepsilon_{i,max}^2 + \frac{1}{2} G_{i,max}^2 L_{\omega_i}^2 a_{i,max}^2}{\lambda_{min}(Q_i)}}. \tag{9}$$

According to the Lyapunov extension theorem, when (9) is satisfied, the weight estimation error $\bar{W}_i$ and the local consensus errors $\bar{e}_i$ are UUB.

Case 2: When it exceeds the threshold conditions learned by Algorithm 1, i.e., $t_i \in \left\{t_i^k, k \in N\right\}$. We consider the Lyapunov function as

$$\begin{aligned} \Delta L_i &= \Delta L_{i1} + \Delta L_{i2} \\ &= V^*(\bar{e}_i(t_i^k)) - V^*(\bar{e}_i) \\ &\quad + \frac{1}{2} tr((\bar{W}_i^k)^\top (\bar{W}_i^k)) - \frac{1}{2} tr(\bar{W}_i^\top \bar{W}_i). \end{aligned} \tag{10}$$

Due to the fact that the system (3.8) is continuous, we can obtain that

$$\Delta L_{i1} = 0. \tag{11}$$

Thus, we have

$$\begin{aligned}
\Delta L_i &= \Delta L_{i2} \\
&= \frac{1}{2}\mathrm{tr}[(\bar{W}_i - \alpha_i \nabla\Phi_i \dot{\bar{e}}_i((\nabla\Phi_i \dot{\bar{e}}_i)^\top \bar{W}_i - \sigma_i))^\top \\
&\times (\bar{W}_i - \alpha_i \nabla\Phi_i \dot{\bar{e}}_i((\nabla\Phi_i \dot{\bar{e}}_i)^\top \bar{W}_i - \sigma_i)) - \bar{W}_i^\top \bar{W}_i] \\
&\leq -\alpha_i (\nabla\Phi_i \dot{\bar{e}}_i)^\top (\nabla\Phi_i \dot{\bar{e}}_i) \left\|\bar{W}_i\right\|^2 + \alpha_i \left\|\bar{W}_i^\top (\nabla\Phi_i \dot{\bar{e}}_i)\sigma_i\right\| \\
&+ \frac{\alpha_i^2}{2}\left\|(\nabla\Phi_i \dot{\bar{e}}_i)(\nabla\Phi_i \dot{\bar{e}}_i)^\top \bar{W}_i - (\nabla\Phi_i \dot{\bar{e}}_i)\sigma_i\right\|^2 \\
&\leq -\alpha_i \varphi_{i,min}^2 \left\|\bar{W}_i\right\|^2 + \alpha_i \varphi_{i,max}\sigma_{i,max}\left\|\bar{W}_i\right\| \\
&+ \frac{\alpha_i^2}{2}\left\|(\nabla\Phi_i \dot{\bar{e}}_i)(\nabla\Phi_i \dot{\bar{e}}_i)^\top \bar{W}_i\right\|^2 + \frac{\alpha_i^2}{2}\left\|(\nabla\Phi_i \dot{\bar{e}}_i)\sigma_i\right\|^2 \\
&+ \alpha_i^2 \left\|(\nabla\Phi_i \dot{\bar{e}}_i)(\nabla\Phi_i \dot{\bar{e}}_i)^\top \bar{W}_i\right\| \left\|(\nabla\Phi_i \dot{\bar{e}}_i)\sigma_i\right\| \\
&\leq (-\alpha_i \varphi_{i,min}^2 + \tfrac{1}{2}\alpha_i^2 \varphi_{i,max}^4)\left\|\bar{W}_i\right\|^2 + \tfrac{1}{2}\alpha_i^2 \varphi_{i,max}^2 \sigma_{i,max}^2 \\
&+ (\alpha_i \varphi_{i,max}\sigma_{i,max} + \alpha_i^2 \varphi_{i,max}^3 \sigma_{i,max})\left\|\bar{W}_i\right\| \\
&\leq X_i \left\|\bar{W}_i\right\|^2 + Y_i \left\|\bar{W}_i\right\| + Z_i,
\end{aligned} \tag{12}$$

in which

$$X_i = -\alpha_i \varphi_{i,min}^2 + \tfrac{1}{2}\alpha_i^2 \varphi_{i,max}^4,$$

$$Y_i = \alpha_i \varphi_{i,max}\sigma_{i,max} + \alpha_i^2 \varphi_{i,max}^3 \sigma_{i,max},$$

and

$$Z_i = \tfrac{1}{2}\alpha_i^2 \varphi_{i,max}^2 \sigma_{i,max}^2 > 0.$$

In order to guarantee $\Delta L_i < 0$, the following two conditions should be satisfied:

$$X_i < 0, \tag{13}$$

and

$$\left\|\bar{W}_i\right\| > \frac{-Y_i + \sqrt{Y_i^2 - 4X_i Z_i}}{2X_i}. \tag{14}$$

Combining case 1 and case 2, we complete the proof.

Appendix B. Proof of Theorem 2

Considering the last trigger instant consensus error when $t \in (t_i^k, t_i^{k+1}]$, which can be expressed as

$$\begin{aligned}
&\left\|\dot{\bar{e}}_i(t_i^k) - \dot{\bar{e}}_i(t)\right\| \\
&= \left\|F_i(\bar{e}_i(t_i^k)) - F_i(\bar{e}_i(t)) + G_i(\bar{e}_i(t_i^k))\hat{\omega}_i - G_i(\bar{e}_i(t))\hat{\omega}_i\right\| \\
&\leq \left\|F_i(\bar{e}_i(t_i^k)) - F_i(\bar{e}_i(t))\right\| + \left\|G_i(\bar{e}_i(t_i^k))\hat{\omega}_i - G_i(\bar{e}_i(t))\hat{\omega}_i\right\| \\
&\leq L_{F_i}\left\|\bar{e}_i(t_i^k) - \bar{e}_i(t)\right\| + 2G_{i,max}\left\|\frac{1}{2}R_{ii}^{-1}G_i^\top\left(\bar{e}_i(t_i^k)\right)\nabla\phi_i^\top \hat{W}_i\right\|.
\end{aligned} \tag{15}$$

Referring to the comparison lemma in [29], one has

$$\begin{aligned}
\left\|\bar{e}_i(t_i^k) - \bar{e}_i(t)\right\| &\leq e^{L_{F_i}(t-t_i^{k+})}\left\|\bar{e}_i(t_i^k) - \bar{e}_i(t_i^{k+})\right\| \\
&+ \frac{1}{2}\int_{t_i^{k+}}^{t} e^{L_{F_i}(t-\tau)}\Pi_i d\tau,
\end{aligned} \tag{16}$$

where $\Pi_i = 2G_{i,max}\left\|\frac{1}{2}R_{ii}^{-1}G_i^\top\left(\bar{e}_i(t_i^k)\right)\nabla\phi_i^\top \hat{W}_i\right\|$ can be inferred to be bounded, i.e., $\|\Pi_i\| \leq \Pi_{i,max}$. Considering that $\left\|\bar{e}_i(t_i^k) - \bar{e}_i(t_i^{k+})\right\| = 0$, we can obtain

$$\begin{aligned}
\left\|\bar{e}_i(t_i^k) - \bar{e}_i(t)\right\| &\leq \frac{\Pi_{i,max}}{2}\int_{t_i^{k+}}^{t} e^{L_{F_i}(t-\tau)}d\tau \\
&\leq \frac{\Pi_{i,max}}{2L_{F_i}}(e^{L_{F_i}(t-t_i^{k+})} - 1), t \in (t_i^k, t_i^{k+1}].
\end{aligned} \tag{17}$$

Due to the event-triggered condition, one has

$$\left\|\bar{e}_i(t) - \bar{e}_i(t_i^k)\right\| \leq a_t(i) \leq a_{i,max}, t \in (t_i^k, t_i^{k+1}). \tag{18}$$

Combining (17) and (18), we can conclude that

$$a_{i,max} \leq \frac{\Pi_{i,max}}{2L_{F_i}}(e^{L_{F_i}(t-t_i^{k+})} - 1), t_i \in (t_i^k, t_i^{k+1}). \tag{19}$$

Through analysis, we find that there exists a non-negative lower bound on the time interval, that is

$$t_i^{k+1} - t_i^k \geq t - t_i^{k+} \geq \frac{1}{L_{F_i}}\log(\frac{2a_{i,max}L_{F_i}}{\Pi_{i,max}} + 1) > 0. \tag{20}$$

Thus, the Zeno behavior can be avoided, which completes our proof.

References

1. M. A. Khamis, W. Gomaa, Adaptive multi-objective reinforcement learning with hybrid exploration for traffic signal control based on cooperative multi-agent framework, *Eng. Appl. Artif. Intell.*, **29** (2014), 134–151. https://doi.org/10.1016/j.engappai.2014.01.007
2. L. Ding, S. Li, H. Gao, C. Chen, Z. Deng, Adaptive partial reinforcement learning neural network-based tracking control for wheeled mobile robotic systems, *IEEE T. Syst. Man Cybern.-Syst.*, **50** (2020), 2512–2523. https://doi.org/10.1109/TSMC.2018.2819191
3. O. P. Mahela, M. Khosravy, N. Gupta, B. Khan, H. H. Alhelou, R. Mahla, et al., Comprehensive overview of multi-agent systems for controlling smart grids, *CSEE J. Power Energy Syst*, **8** (2022), 115–131. https://doi.org/10.17775/CSEEJPES.2020.03390

4. J. Wang, J. Gao, P. Wu, Attack-resilient event-triggered formation control of multi-agent systems under periodic DoS attacks using complex Laplacian, *ISA Trans.*, **128** (2022), 10–16. https://doi.org/10.1016/j.isatra.2021.10.030
5. H. Zhang, H. Jiang, Y. Luo, G. Xiao, Data-driven optimal consensus control for discrete-time multi-agent systems with unknown dynamics using reinforcement learning method, *IEEE Trans. Ind. Electron.*, **64** (2017), 4091–4100. https://doi.org/10.1109/TIE.2016.2542134
6. Q. Wei, D. Liu, H. Lin, Value iteration adaptive dynamic programming for optimal control of discrete-time nonlinear systems, *IEEE Trans. Cybern.*, **46** (2016), 840–853. https://doi.org/10.1109/TCYB.2015.2492242
7. R. Song, F. L. Lewis, Q. Wei, H. Zhang, Off-policy actor-critic structure for optimal control of unknown systems with disturbances, *IEEE Trans. Cybern.*, **46** (2016), 1041–1050. https://doi.org/10.1109/TCYB.2015.2421338
8. H. Lyu, Y. Lyu, Y. Gao, H. Qian, S. Du, MIMO fuzzy adaptive control systems based on fuzzy semi-tensor product, *Math. Model. Control*, **3** (2023), 316–330. https://doi.org/10.3934/mmc.2023026
9. V. Stojanovic, Fault-tolerant control of a hydraulic servo actuator via adaptive dynamic programming, *Math. Model. Control*, **3** (2023), 181–191. https://doi.org/10.3934/mmc.2023016
10. J. Zhang, H. Zhang, T. Feng, Distributed optimal consensus control for nonlinear multiagent system with unknown dynamic, *IEEE T. Neural Networks Learn. Syst.*, **29** (2018), 3339–3348. https://doi.org/10.1109/TNNLS.2017.2728622
11. S. Jiao, Q. Wei, A new optimal consensus control for nonlinear multi-agent systems, *2023 9th International Conference on Control Science and Systems Engineering (ICCSSE)*, 2023, 1–6. https://doi.org/10.1109/iccsse59359.2023.10244873
12. B. Zhao, D. Liu, C. Luo, Reinforcement learning-based optimal stabilization for unknown nonlinear systems subject to inputs with uncertain constraints, *IEEE T. Neural Networks Learn. Syst.*, **31** (2020), 4330–4340. https://doi.org/10.1109/TNNLS.2019.2954983
13. Q. Wei, R. Song, P. Yan, Data-driven zero-sum neuro-optimal control for a class of continuous-time unknown nonlinear systems with disturbance using ADP, *IEEE T. Neural Networks Learn. Syst.*, **27** (2016), 444–458. https://doi.org/10.1109/TNNLS.2015.2464080
14. X. Zhong, H. He, An event-triggered ADP control approach for continuous-time system With unknown internal states, *IEEE Trans. Cybern.*, **47** (2017), 683–694. https://doi.org/10.1109/TCYB.2016.2523878
15. B. Li, N. Chen, B. Luo, J. Chen, C. Yang, W. Gui, ADP-based event-triggered constrained optimal control on spatiotemporal process: application to temperature field in roller kiln, *IEEE T. Neural Networks Learn. Syst.*, **35** (2024), 3229–3241. https://doi.org/10.1109/TNNLS.2023.3267516
16. R. Mu, A. Wei, H. Li, X. Zhang, Y. Qi, Bipartite output consensus for heterogeneous multi-agent systems with observer-based adaptive event-triggered strategy, *IEEE Syst. J.*, **17** (2023), 6011–6021. https://doi.org/10.1109/JSYST.2023.3307686
17. L. Chen, F. Hao, Dynamic event-triggered robust stabilization of continuous-time nonaffine nonlinear systems based on ADP, *2023 42nd Chinese Control Conference (CCC)*, 2023, 1611–1616. https://doi.org/10.23919/CCC58697.2023.10240133
18. A. Sahoo, H. Xu, S. Jagannathan, Near optimal event-triggered control of nonlinear discrete-time systems using neurodynamic programming, *IEEE T. Neural Networks Learn. Syst.*, **27** (2016), 1801–1815. https://doi.org/10.1109/TNNLS.2015.2453320
19. Z. Wang, Q. Wei, D. Liu, Y. Luo, Event-triggered adaptive control for discrete-time zero-sum games, *2019 International Joint Conference on Neural Networks (IJCNN)*, 2019, 1–7. https://doi.org/10.1109/IJCNN.2019.8852115

20. L. Dong, X. Zhong, C. Sun, H. He, Adaptive event-triggered control based on heuristic dynamic programming for nonlinear discrete-time systems, *IEEE T. Neural Networks Learn. Syst.*, **28** (2017), 1594–1605. https://doi.org/10.1109/TNNLS.2016.2541020

21. B. Xu, Y. X. Li, Z. Hou, C. K. Ahn, Dynamic event-triggered reinforcement learning-based consensus tracking of nonlinear multi-agent systems, *IEEE Trans. Circuits Syst.-I*, **70** (2023), 2120–2132. https://doi.org/10.1109/TCSI.2023.3246001

22. G. Wen, C. L. P. Chen, J. Feng, N. Zhou, Optimized multi-agent formation control based on an identifier-Actor-Critic reinforcement learning algorithm, *IEEE Trans. Fuzzy Syst.*, **26** (2018), 2719–2731. https://doi.org/10.1109/TFUZZ.2017.2787561

23. J. Peng, C. Mu, K. Wang, A nearly optimal multi-agent formation control with reinforcement learning, *2021 40th Chinese Control Conference (CCC)*, 2021, 5315–5320. https://doi.org/10.23919/CCC52363.2021.9550415

24. G. Wen, C. L. P. Chen, Optimized backstepping consensus control using reinforcement learning for a class of nonlinear strict-feedback-dynamic multi-agent systems, *IEEE T. Neural Networks Learn. Syst.*, **34** (2023), 1524–1536. https://doi.org/10.1109/TNNLS.2021.3105548

25. G. Wen, B. Li, Optimized leader-follower consensus control using reinforcement learning for a class of second-order nonlinear multiagent systems, *IEEE T. Syst. Man Cybern. Syst.*, **52** (2022), 5546–5555. https://doi.org/10.1109/TSMC.2021.3130070

26. M. Long, H. Su, Z. Zeng, Model-free event-triggered consensus algorithm for multiagent systems using reinforcement learning method, *IEEE T. Syst. Man Cybern. Syst.*, **52** (2022), 5212–5221. https://doi.org/10.1109/TSMC.2021.3120008

27. P. Chen, S. Liu, D. Zhang, A Q-learning based dynamic event-triggered control for load frequency regulation of power systems with denial-of-service attacks, *2021 IEEE 30th International Symposium on Industrial Electronics (ISIE)*, 2021, 1–5. https://doi.org/10.1109/ISIE45552.2021.9576200

28. S. Wang, X. Jin, S. Mao, A. V. Vasilakos, Y. Tang, Model-free event-triggered optimal consensus control of multiple euler-lagrange systems via reinforcement learning, *IEEE Trans. Network Sci. Eng.*, **8** (2021), 246–258. https://doi.org/10.1109/TNSE.2020.3036604

29. H. K. Khalil, *Nonlinear systems*, 3 Eds., Upper Saddle River: Prentice Hall, 2002.

AIMS Press