Frontiers of Information Technology & Electronic Engineering
www.jzus.zju.edu.cn; engineering.cae.cn; www.springerlink.com
ISSN 2095-9184 (print); ISSN 2095-9230 (online)
E-mail: jzus@zju.edu.cn

Optimal synchronization control for multi-agent systems with input saturation: a nonzero-sum game*

Hongyang LI[1,2], Qinglai WEI‡[1,2,3]

[1]*School of Artificial Intelligence, University of Chinese Academy of Sciences, Beijing 100049, China*

[2]*The State Key Laboratory for Management and Control of Complex Systems, Institute of Automation, Chinese Academy of Sciences, Beijing 100190, China*

[3]*Institute of Systems Engineering, Macau University of Science and Technology, Macau 999078, China*

E-mail: lihongyang2019@ia.ac.cn; qinglai.wei@ia.ac.cn

Received Jan. 14, 2022; Revision accepted Mar. 7, 2022; Crosschecked Apr. 29, 2022; Published online May 20, 2022

Abstract: This paper presents a novel optimal synchronization control method for multi-agent systems with input saturation. The multi-agent game theory is introduced to transform the optimal synchronization control problem into a multi-agent nonzero-sum game. Then, the Nash equilibrium can be achieved by solving the coupled Hamilton–Jacobi–Bellman (HJB) equations with nonquadratic input energy terms. A novel off-policy reinforcement learning method is presented to obtain the Nash equilibrium solution without the system models, and the critic neural networks (NNs) and actor NNs are introduced to implement the presented method. Theoretical analysis is provided, which shows that the iterative control laws converge to the Nash equilibrium. Simulation results show the good performance of the presented method.

Key words: Optimal synchronization control; Multi-agent systems; Nonzero-sum game; Adaptive dynamic programming; Input saturation; Off-policy reinforcement learning; Policy iteration

https://doi.org/10.1631/FITEE.2200010 **CLC number:** TP13

1 Introduction

Multi-agent synchronization control has attracted much attention due to its high efficiency and computational performance (Wieland et al., 2011; Wei et al., 2018, 2020, 2021; Li JQ et al., 2021; Rehák and Lynnyk, 2021; Zhang KQ et al., 2021). Generally speaking, synchronization control problems require that the agents synthesize to the same value (Cao et al., 2015; Garcia et al., 2017; Yang JY et al., 2019) or track the trajectories of leaders (Du et al., 2014; Zhao et al., 2014) by designing distributed control laws. Because of the practical significance of multi-agent systems, many researchers have devoted themselves to tackling various synchronization control problems, including switching topologies (Thunberg et al., 2014), system faults (Ma and Yang, 2016) and so on (Han et al., 2013; Wei et al., 2015; He et al., 2018). Among the research works on synchronization control, optimal synchronization control, which requires each agent to minimize its own local performance index function, is a promising research direction. Multi-agent cooperative games provide an effective tool to study multi-agent optimal control problems, and they rely on solving coupled Hamilton–Jacobi (HJ) equations (Vamvoudakis et al., 2012). However, coupled HJ equations are hard to solve, which limits the applications of cooperative game theory in synchronization control problems.

Reinforcement learning is an effective method

‡ Corresponding author

* Project supported by the National Key R&D Program of China (No. 2018YFB1702300) and the National Natural Science Foundation of China (Nos. 61722312 and 61533017)

ORCID: Hongyang LI, https://orcid.org/0000-0001-5891-134X; Qinglai WEI, https://orcid.org/0000-0001-7002-9800

for solving coupled HJ equations. The main idea of reinforcement learning is to solve the coupled HJ equations forward by time, which can reduce the computational burden (Wang et al., 2009; Wei and Liu, 2014; Wei et al., 2014, 2016, 2017; Zhang HG et al., 2015; Yang N et al., 2019; Zhang LD et al., 2019). In recent years, reinforcement learning has been further developed to solve multi-agent cooperative game problems. Vamvoudakis et al. (2012) proposed an online policy iteration method for optimal synchronization control problems; however, the external disturbance was not considered. In Jiao et al. (2016), a novel policy iteration method was proposed for the multi-agent zero-sum game problem, and disturbance rejection was achieved. In Wei et al. (2015), the graphical game was studied for heterogeneous multi-agent systems. The off-policy reinforcement learning method was proposed to solve multi-agent synchronization control problems by Li JN et al. (2017), and the input constraint was considered by Qin et al. (2019). However, there are few research results considering cooperative game problems with input saturation, which motivates our study.

In this paper, the multi-agent optimal synchronization control problem with input saturation is studied based on cooperative game theory and reinforcement learning. Compared with Qin et al. (2019), we consider coupled terms with neighboring agents in the performance index functions. The main contributions can be summarized as follows:

1. A novel off-policy reinforcement learning method is presented without the information of system models for cooperative game problems of multi-agent systems. The control constraint and coupled terms in the performance index functions are considered which broaden the application scope of the presented method.

2. The characteristics of the presented model-free off-policy reinforcement learning method, including convergence and optimality, are analyzed, showing that the solutions obtained from the presented method converge to the Nash equilibrium.

3. Critic neural networks (NNs) and actor NNs are used to implement the off-policy reinforcement learning algorithm. Simulation results verify the good performance of the presented method.

2 Problem formulation

2.1 Graph theory

Let $G_r = (V, \varepsilon, \boldsymbol{E})$ be a directed graph, where $V = \{v_1, v_2, \ldots, v_N\}$ denotes the nonempty finite vertex set. Furthermore, $\varepsilon \subseteq V \times V$ is the set of edges. An edge of graph G_r is denoted as ε_{ij}, which means that agent j is a neighbor of agent i. $\boldsymbol{E} = [e_{ij}] \in \mathbb{R}^{N\times N}$ is the adjacency matrix, where e_{ij} represents the weight of edge ε_{ij}. If $\varepsilon_{ij} \in \varepsilon$, $e_{ij} > 0$; otherwise, $e_{ij} = 0$. Let the set of neighbors of agent i be $N_i = \{v_j | (v_j, v_i) \in \varepsilon\}$. Define $\boldsymbol{G} = \mathrm{diag}(g_i) \in \mathbb{R}^{N\times N}$ as the pinning matrix. If agent i has access to the leader, $g_i > 0$; otherwise, $g_i = 0$. Define the Laplacian matrix as $\boldsymbol{L} = \boldsymbol{D} - \boldsymbol{E}$, where $\boldsymbol{D} = \mathrm{diag}(d_i)$ and $d_i = \sum_j e_{ij}$.

2.2 Multi-agent synchronization control

For $i = 1, 2, \ldots, N$, consider the following systems:

$$\dot{\boldsymbol{x}}_i = \boldsymbol{A}\boldsymbol{x}_i + \boldsymbol{B}\boldsymbol{u}_i, \tag{1}$$

where $\boldsymbol{x}_i \in \mathbb{R}^n$ and $\boldsymbol{u}_i \in \mathcal{U}_i \subset \mathbb{R}^m$ are the system state and control, respectively. Here, $\boldsymbol{A}$ and $\boldsymbol{B}$ are system matrices with suitable dimensions. $\mathcal{U}_i = \{\boldsymbol{u}_i | \boldsymbol{u}_i \in \mathbb{R}^m, \|\boldsymbol{u}_i\|_\infty \le \lambda_i\}$ ($\lambda_i > 0$ is a known constant). Let the leader dynamics be

$$\dot{\boldsymbol{x}}_0 = \boldsymbol{A}\boldsymbol{x}_0, \tag{2}$$

where $\boldsymbol{x}_0 \in \mathbb{R}^n$ is the system state. Then, we can define the synchronization error as

$$\boldsymbol{\delta}_i = \sum_{j\in N_i} e_{ij}(\boldsymbol{x}_i - \boldsymbol{x}_j) + g_i(\boldsymbol{x}_i - \boldsymbol{x}_0). \tag{3}$$

Taking the derivative of Eq. (3), we have

$$\dot{\boldsymbol{\delta}}_i = \boldsymbol{A}\boldsymbol{\delta}_i + (d_i + g_i)\boldsymbol{B}\boldsymbol{u}_i - \sum_{j\in N_i} e_{ij}\boldsymbol{B}\boldsymbol{u}_j. \tag{4}$$

For system (4), the performance index function can be given as

$$J_i(\boldsymbol{\delta}_i(0), \boldsymbol{u}_i, \boldsymbol{u}_{-i}) = \int_0^\infty \left(\boldsymbol{\delta}_i^{\mathrm{T}}\boldsymbol{Q}_{ii}\boldsymbol{\delta}_i + R_i(\boldsymbol{u}_i) + \sum_{j\in N_i} R_i(\boldsymbol{u}_j)\right)\mathrm{d}t, \tag{5}$$

where the term $\boldsymbol{u}_{-i}$ represents the policies of the neighbors of agent i. $\boldsymbol{Q}_{ii}>0$,

$$R_i(\boldsymbol{u}_i)=2\int_{\boldsymbol{0}}^{\boldsymbol{u}_i}\left(\lambda_i\Psi^{-1}\left(\boldsymbol{t}/\lambda_i\right)\right)^{\mathrm{T}}\boldsymbol{R}_{ii}\mathrm{d}\boldsymbol{t},$$
$$R_i(\boldsymbol{u}_j)=2\int_{\boldsymbol{0}}^{\boldsymbol{u}_j}\left(\lambda_j\Psi^{-1}\left(\boldsymbol{t}/\lambda_j\right)\right)^{\mathrm{T}}\boldsymbol{R}_{ij}\mathrm{d}\boldsymbol{t},$$

where

$$\boldsymbol{t}=[t_1,t_2,\ldots,t_m]^{\mathrm{T}}\in\mathbb{R}^m,$$
$$\boldsymbol{R}_{ii}=\mathrm{diag}\left(r_{ii,1},r_{ii,2},\ldots,r_{ii,m}\right)>0,$$
$$\boldsymbol{R}_{ij}=\mathrm{diag}\left(r_{ij,1},r_{ij,2},\ldots,r_{ij,m}\right)\geq 0,$$

and Ψ^{-1} is the inverse function of the hyperbolic tangent function (i.e., $\Psi^{-1}=\mathrm{arctanh}(\cdot)$, or equivalently, $\Psi(\cdot)\triangleq\tanh(\cdot)$). Then, $R_i(\boldsymbol{u}_i)$ and $R_i(\boldsymbol{u}_j)$ can be written as

$$\begin{aligned}R_i(\boldsymbol{u}_i)&=2\int_{\boldsymbol{0}}^{\boldsymbol{u}_i}\left(\lambda_i\Psi^{-1}\left(\boldsymbol{t}/\lambda_i\right)\right)^{\mathrm{T}}\boldsymbol{R}_{ii}\mathrm{d}\boldsymbol{t}\\&=2\sum_{q=1}^{m}r_{ii,q}\int_0^{u_{iq}}\lambda_i\Psi^{-1}\left(t_q/\lambda_i\right)\mathrm{d}t_q,\end{aligned}\tag{6}$$

$$\begin{aligned}R_i(\boldsymbol{u}_j)&=2\int_{\boldsymbol{0}}^{\boldsymbol{u}_j}\left(\lambda_j\Psi^{-1}\left(\boldsymbol{t}/\lambda_j\right)\right)^{\mathrm{T}}\boldsymbol{R}_{ij}\mathrm{d}\boldsymbol{t}\\&=2\sum_{q=1}^{m}r_{ij,q}\int_0^{u_{jq}}\lambda_j\Psi^{-1}\left(t_q/\lambda_j\right)\mathrm{d}t_q.\end{aligned}\tag{7}$$

For Eq. (5), the Nash equilibrium condition can be described as

$$J_i(\boldsymbol{u}_i^*,\boldsymbol{u}_{-i}^*)\leq J_i(\boldsymbol{u}_i,\boldsymbol{u}_{-i}^*),\ i=1,2,\ldots,N.\tag{8}$$

For agent i, we define the iterative value function as

$$\begin{aligned}&V_i\left(\boldsymbol{\delta}_i(t)\right)\\&=\int_t^{\infty}\left(\boldsymbol{\delta}_i^{\mathrm{T}}\boldsymbol{Q}_{ii}\boldsymbol{\delta}_i+R_i(\boldsymbol{u}_i)+\sum_{j\in N_i}R_i(\boldsymbol{u}_j)\right)\mathrm{d}\tau.\end{aligned}\tag{9}$$

Then, we can obtain the Bellman equation as

$$\begin{aligned}&H_i\left(\boldsymbol{\delta}_i,\frac{\partial V_i}{\partial\boldsymbol{\delta}_i},\boldsymbol{u}_i,\boldsymbol{u}_{-i}\right)\\&=\left(\frac{\partial V_i}{\partial\boldsymbol{\delta}_i}\right)^{\mathrm{T}}\left(\boldsymbol{A}\boldsymbol{\delta}_i+\left(d_i+g_i\right)\boldsymbol{B}\boldsymbol{u}_i-\sum_{j\in N_i}e_{ij}\boldsymbol{B}\boldsymbol{u}_j\right)\\&\quad+\boldsymbol{\delta}_i^{\mathrm{T}}\boldsymbol{Q}_{ii}\boldsymbol{\delta}_i+R_i(\boldsymbol{u}_i)+\sum_{j\in N_i}R_i(\boldsymbol{u}_j)\end{aligned}\tag{10}$$

with $V_i(\boldsymbol{0})=0$. According to the stationary condition (Bertsekas, 2007), it can be derived that

$$\begin{aligned}&\frac{\partial H_i}{\partial\boldsymbol{u}_i}=\boldsymbol{0}\Rightarrow\\&\boldsymbol{u}_i^*=-\lambda_i\Psi\left(\frac{1}{2\lambda_i}\left(d_i+g_i\right)\boldsymbol{R}_{ii}^{-1}\boldsymbol{B}^{\mathrm{T}}\frac{\partial V_i^*}{\partial\boldsymbol{\delta}_i}\right).\end{aligned}\tag{11}$$

Substituting Eq. (11) into Eq. (10), we can obtain the Hamilton–Jacobi–Bellman (HJB) equation as

$$\begin{aligned}&\left(\frac{\partial V_i^*}{\partial\boldsymbol{\delta}_i}\right)^{\mathrm{T}}\left(\boldsymbol{A}\boldsymbol{\delta}_i-\lambda_i(d_i+g_i)\boldsymbol{B}\Psi(\boldsymbol{\Delta}_i^*)\right.\\&\left.+\sum_{j\in N_i}\lambda_j e_{ij}\boldsymbol{B}\Psi(\boldsymbol{\Delta}_i^*)\right)+\boldsymbol{\delta}_i^{\mathrm{T}}\boldsymbol{Q}_{ii}\boldsymbol{\delta}_i+R_i(-\lambda_i\Psi(\boldsymbol{\Delta}_i^*))\\&+\sum_{j\in N_i}R_i(-\lambda_j\Psi(\boldsymbol{\Delta}_j^*))=0\end{aligned}\tag{12}$$

with $V_i(\boldsymbol{0})=0$, where

$$\boldsymbol{\Delta}_i^*=\frac{1}{2\lambda_i}\left(d_i+g_i\right)\boldsymbol{R}_{ii}^{-1}\boldsymbol{B}^{\mathrm{T}}\frac{\partial V_i^*}{\partial\boldsymbol{\delta}_i}.$$

We would like to design $\boldsymbol{u}_i$, such that the Nash equilibrium condition represented in inequality (8) and the following state synchronization condition are satisfied:

$$\lim_{t\to\infty}\|\boldsymbol{x}_i-\boldsymbol{x}_0\|=0,\ i=1,2,\ldots,N.\tag{13}$$

Remark 1 The performance index function considered by Qin et al. (2019) is defined as

$$\begin{aligned}&J_i\left(\boldsymbol{\delta}_i(0),\boldsymbol{u}_i,\boldsymbol{u}_{-i}\right)\\&=\int_0^{\infty}\left(\boldsymbol{\delta}_i^{\mathrm{T}}\boldsymbol{Q}_{ii}\boldsymbol{\delta}_i+R_i(\boldsymbol{u}_i)\right)\mathrm{d}t.\end{aligned}\tag{14}$$

Comparing Eqs. (5) and (14), it can be seen that the coupled terms with neighboring agents in the performance index function are considered in this study. In multi-agent systems, the behavior of agent i may have an impact on its neighboring agents. Therefore, the performance index function (5) is more natural for the optimal synchronization control of multi-agent systems.

Remark 2 Based on Eq. (3), it can be derived that

$$\boldsymbol{\delta}=\left((\boldsymbol{L}+\boldsymbol{G})\otimes\boldsymbol{I}_n\right)\left(\boldsymbol{x}-\bar{\boldsymbol{x}}_0\right),$$

where

$$\boldsymbol{\delta}=\left[\boldsymbol{\delta}_1^{\mathrm{T}},\boldsymbol{\delta}_2^{\mathrm{T}},\ldots,\boldsymbol{\delta}_N^{\mathrm{T}}\right]^{\mathrm{T}},$$
$$\boldsymbol{x}=\left[\boldsymbol{x}_1^{\mathrm{T}},\boldsymbol{x}_2^{\mathrm{T}},\ldots,\boldsymbol{x}_N^{\mathrm{T}}\right]^{\mathrm{T}},$$
$$\bar{\boldsymbol{x}}_0=\left[\boldsymbol{x}_0^{\mathrm{T}},\boldsymbol{x}_0^{\mathrm{T}},\ldots,\boldsymbol{x}_0^{\mathrm{T}}\right]^{\mathrm{T}},$$

"$\otimes$" is the Kronecker product, and $\boldsymbol{I}_n$ is an identity matrix with dimension n. According to Vamvoudakis et al. (2012), we have

$$\|\boldsymbol{x}-\bar{\boldsymbol{x}}_0\| \leq \|\boldsymbol{\delta}\|/\sigma_{\min}(\boldsymbol{L}+\boldsymbol{G}),$$

where $\sigma_{\min}(\cdot)$ represents the minimum singular value of the matrix. Therefore, the stability of the tracking error dynamics represented in Eq. (4) guarantees the state synchronization condition denoted by Eq. (13).

3 Main results

3.1 Multi-agent nonzero-sum game

A theorem is provided, which shows that the solution to the HJB equation (i.e., Eq. (12)) satisfies the Nash equilibrium condition (i.e., inequality (8)) under certain conditions.

Theorem 1 Assume that the optimal control law $\boldsymbol{u}_i^*$ is given as shown in Eq. (11), and that V_i is the positive definite smooth solution to the HJB equation (12). Then, system (4) is asymptotically stable. The optimal control laws $\boldsymbol{u}_i^*$ $(i=1,2,\ldots,N)$ constitute the Nash equilibrium, and the solution V_i to the HJB equation (12) is the optimal value of the game, i.e.,

$$J_i^*\left(\boldsymbol{\delta}_i(0),\boldsymbol{u}_i^*,\boldsymbol{u}_{-i}^*\right)=V_i\left(\boldsymbol{\delta}_i(0)\right). \tag{15}$$

Proof Choosing the iterative value function V_i as the Lyapunov function, it can be derived that

$$\frac{\mathrm{d}V_i}{\mathrm{d}t}=\left(\frac{\partial V_i}{\partial \boldsymbol{\delta}_i}\right)^{\mathrm{T}}\left(\boldsymbol{A}\boldsymbol{\delta}_i+(d_i+g_i)\,\boldsymbol{B}\boldsymbol{u}_i-\sum_{j\in N_i}e_{ij}\boldsymbol{B}\boldsymbol{u}_j\right). \tag{16}$$

For the right-hand side of the Bellman equation, i.e., Eq. (10), adding and subtracting $R_i(\boldsymbol{u}_i^*)+\sum_{j\in N_i}R_i(\boldsymbol{u}_j^*)$ and $\left(\frac{\partial V_i}{\partial \boldsymbol{\delta}_i}\right)^{\mathrm{T}}\left(\boldsymbol{A}\boldsymbol{\delta}_i+(d_i+g_i)\,\boldsymbol{B}\boldsymbol{u}_i^*-\sum_{j\in N_i}e_{ij}\boldsymbol{B}\boldsymbol{u}_j^*\right)$, it can be derived that

$$\begin{aligned}
&H_i\left(\boldsymbol{\delta}_i,\frac{\partial V_i}{\partial \boldsymbol{\delta}_i},\boldsymbol{u}_i,\boldsymbol{u}_{-i}\right)\\
=&H_i\left(\boldsymbol{\delta}_i,\frac{\partial V_i}{\partial \boldsymbol{\delta}_i},\boldsymbol{u}_i^*,\boldsymbol{u}_{-i}^*\right)+R_i(\boldsymbol{u}_i)-R_i(\boldsymbol{u}_i^*)\\
&+\sum_{j\in N_i}\left(R_i(\boldsymbol{u}_j)-R_i(\boldsymbol{u}_j^*)\right)\\
&+\left(\frac{\partial V_i}{\partial \boldsymbol{\delta}_i}\right)^{\mathrm{T}}(d_i+g_i)\,\boldsymbol{B}\left(\boldsymbol{u}_i-\boldsymbol{u}_i^*\right)\\
&-\left(\frac{\partial V_i}{\partial \boldsymbol{\delta}_i}\right)^{\mathrm{T}}\sum_{j\in N_i}e_{ij}\boldsymbol{B}\left(\boldsymbol{u}_j-\boldsymbol{u}_j^*\right).
\end{aligned} \tag{17}$$

According to Eq. (12), we have

$$H_i\left(\boldsymbol{\delta}_i,\frac{\partial V_i}{\partial \boldsymbol{\delta}_i},\boldsymbol{u}_i^*,\boldsymbol{u}_{-i}^*\right)=0.$$

Letting $\boldsymbol{u}_i=\boldsymbol{u}_i^*$ and $\boldsymbol{u}_j=\boldsymbol{u}_j^*$ $(j\in N_i)$, Eq. (17) can be written as

$$\frac{\mathrm{d}V_i}{\mathrm{d}t}+\boldsymbol{\delta}_i^{\mathrm{T}}\boldsymbol{Q}_{ii}\boldsymbol{\delta}_i+R_i(\boldsymbol{u}_i)+\sum_{j\in N_i}R_i(\boldsymbol{u}_j)=0. \tag{18}$$

It can be derived that $\frac{\mathrm{d}V_i}{\mathrm{d}t}<0$, and thus system (4) is asymptotically stable.

Based on Eqs. (5) and (9), it can be derived that

$$\begin{aligned}
&J_i\left(\boldsymbol{\delta}_i(0),\boldsymbol{u}_i,\boldsymbol{u}_{-i}\right)\\
=&\int_0^\infty\left(\boldsymbol{\delta}_i^{\mathrm{T}}\boldsymbol{Q}_{ii}\boldsymbol{\delta}_i+R_i(\boldsymbol{u}_i)+\sum_{j\in N_i}R_i(\boldsymbol{u}_j)\right)\mathrm{d}t\\
&+V_i\left(\boldsymbol{\delta}_i(0)\right)-V_i\left(\boldsymbol{\delta}_i(\infty)\right)\\
&+\int_0^\infty\left(\frac{\partial V_i}{\partial \boldsymbol{\delta}_i}\right)^{\mathrm{T}}\left(\boldsymbol{A}\boldsymbol{\delta}_i+(d_i+g_i)\,\boldsymbol{B}\boldsymbol{u}_i\right.\\
&\left.-\sum_{j\in N_i}e_{ij}\boldsymbol{B}\boldsymbol{u}_j\right)\mathrm{d}t.
\end{aligned} \tag{19}$$

Because of the asymptotic stability of system (4) and the boundary condition $V_i(\boldsymbol{0})=0$, it can be derived that $V_i(\boldsymbol{\delta}_i(\infty))=0$. Then, substituting Eq. (12) into Eq. (19) and completing the squares, we can obtain

$$\begin{aligned}
&J_i(\boldsymbol{\delta}_i(0),\boldsymbol{u}_i,\boldsymbol{u}_{-i})\\
=&V_i(\boldsymbol{\delta}_i(0))+\int_0^\infty\left(2\int_{\boldsymbol{u}_i^*/\lambda_i}^{\boldsymbol{u}_i/\lambda_i}\lambda_i^2(\Psi^{-1}(\boldsymbol{s})\right.\\
&-\Psi^{-1}(\boldsymbol{u}_i^*/\lambda_i))^{\mathrm{T}}\boldsymbol{R}_{ii}\mathrm{d}\boldsymbol{s}\\
&+2\sum_{j\in N_i}\int_{\boldsymbol{u}_j^*/\lambda_j}^{\boldsymbol{u}_j/\lambda_j}\lambda_j^2(\Psi^{-1}(\boldsymbol{s}))^{\mathrm{T}}\boldsymbol{R}_{ij}\mathrm{d}\boldsymbol{s}\\
&\left.-\left(\frac{\partial V_i}{\partial \boldsymbol{\delta}_i}\right)^{\mathrm{T}}\sum_{j\in N_i}e_{ij}\boldsymbol{B}\left(\boldsymbol{u}_j-\boldsymbol{u}_j^*\right)\right)\mathrm{d}t.
\end{aligned} \tag{20}$$

For $\boldsymbol{u}_{-i}=\boldsymbol{u}_{-i}^*$, it can be obtained that

$$\begin{aligned}
&J_i\left(\boldsymbol{\delta}_i(0),\boldsymbol{u}_i,\boldsymbol{u}_{-i}^*\right)\\
=&V_i\left(\boldsymbol{\delta}_i(0)\right)+\int_0^\infty 2\lambda_i^2\int_{\boldsymbol{u}_i^*/\lambda_i}^{\boldsymbol{u}_i/\lambda_i}\left(\Psi^{-1}\left(\boldsymbol{s}\right)\right.\\
&\left.-\Psi^{-1}\left(\boldsymbol{u}_i^*/\lambda_i\right)\right)^{\mathrm{T}}\boldsymbol{R}_{ii}\mathrm{d}\boldsymbol{s}\mathrm{d}t.
\end{aligned} \tag{21}$$

For Eq. (21), it can be derived that

$$\begin{aligned}&\int_{\boldsymbol{u}_i^*/\lambda_i}^{\boldsymbol{u}_i/\lambda_i}\left(\Psi^{-1}(\boldsymbol{s})-\Psi^{-1}(\boldsymbol{u}_i^*/\lambda_i)\right)^{\mathrm{T}}\boldsymbol{R}_{ii}\mathrm{d}\boldsymbol{s}\\=&\sum_{q=1}^{m}r_{ii,q}\int_{u_{iq}^*/\lambda_i}^{u_{iq}/\lambda_i}\left(\Psi^{-1}(s_q)-\Psi^{-1}\left(u_{iq}^*/\lambda_i\right)\right)\mathrm{d}s_q,\end{aligned}\tag{22}$$

where $\Psi^{-1}=\operatorname{arctanh}(\cdot)$ is monotonically increasing, i.e., $\left(\Psi^{-1}\right)'>0$. Therefore, based on the mean value theorem for integrals, it can be derived that

$$\begin{aligned}&\int_{u_{iq}^*/\lambda_i}^{u_{iq}/\lambda_i}\left(\Psi^{-1}(s_q)-\Psi^{-1}\left(u_{iq}^*/\lambda_i\right)\right)\mathrm{d}s_q\\&=\left(\Psi^{-1}(\bar{s}_q)-\Psi^{-1}\left(u_{iq}^*/\lambda_i\right)\right)\left(u_{iq}/\lambda_i-u_{iq}^*/\lambda_i\right)\\&>0,\end{aligned}\tag{23}$$

where $\bar{s}_q$ is between u_{iq}^*/λ_i and u_{iq}/λ_i. Therefore, the second term of the right-hand side of Eq. (21) is positive. If $\boldsymbol{u}_i=\boldsymbol{u}_i^*$, it can be derived that the solution V_i to Eq. (12) is the optimal value of the game. Comparing Eqs. (21) and (15), it can be obtained that the condition expressed in inequality (8) is satisfied. The proof is completed.

3.2 Policy iteration method for solving the HJB equation

In the previous subsection, it was derived that the optimal control, represented in Eq. (11), can be calculated to construct the Nash equilibrium represented in inequality (8). However, the optimal control in Eq. (11) requires the information of V_i^*, which can be calculated from the HJB equation (12). The HJB equation (12) is a nonlinear partial differential equation, which is hard to solve analytically. Therefore, a policy iteration method is provided (Algorithm 1) to solve the HJB equation (12) numerically. Then, a theorem can be provided, which shows the convergence of the presented policy iteration method.

Theorem 2 Assume that agent i and its neighbors update their control policies according to Algorithm 1. Then, policies $\boldsymbol{u}_i^k$ and $\boldsymbol{u}_{-i}^k$ converge to the Nash equilibrium, and V_i^k converges to V_i^*, where V_i^* is the solution to the HJB equation (12).

Proof Integrating $\dot{V}_i^k-\dot{V}_i^{k+1}$ along the system

$$\dot{\boldsymbol{\delta}}_i=\boldsymbol{A}\boldsymbol{\delta}_i+(d_i+g_i)\boldsymbol{B}\boldsymbol{u}_i^{k+1}-\sum_{j\in N_i}e_{ij}\boldsymbol{B}\boldsymbol{u}_j^k,\tag{26}$$

Algorithm 1 Policy iteration method

Initialization: Give the initial stabilizing control laws $\boldsymbol{u}_i^0, i=1,2,\ldots,N$

Iteration:

1: Let $k=0$

2: For $i=1,2,\ldots,N$, solve for V_i^k from the following equation:

$$\begin{aligned}&\left(\frac{\partial V_i^k}{\partial\boldsymbol{\delta}_i}\right)^{\mathrm{T}}\left(\boldsymbol{A}\boldsymbol{\delta}_i+(d_i+g_i)\boldsymbol{B}\boldsymbol{u}_i^k-\sum_{j\in N_i}e_{ij}\boldsymbol{B}\boldsymbol{u}_j^k\right)\\&+\boldsymbol{\delta}_i^{\mathrm{T}}\boldsymbol{Q}_{ii}\boldsymbol{\delta}_i+R_i(\boldsymbol{u}_i^k)+\sum_{j\in N_i}R_i(\boldsymbol{u}_j^k)=0\end{aligned}\tag{24}$$

3: For $i=1,2,\ldots,N$, update the control laws as

$$\boldsymbol{u}_i^{k+1}=-\lambda_i\Psi\left(\frac{1}{2\lambda_i}(d_i+g_i)\boldsymbol{R}_{ii}^{-1}\boldsymbol{B}^{\mathrm{T}}\frac{\partial V_i^k}{\partial\boldsymbol{\delta}_i}\right)\tag{25}$$

4: Let $k=k+1$; repeat steps 2 and 3 until convergence

5: **return** V_i^k, $\boldsymbol{u}_i^k$, $i=1,2,\ldots,N$

we have

$$\begin{aligned}&V_i^k-V_i^{k+1}\\&=\int_t^{\infty}\left(\left(\frac{\partial V_i^{k+1}}{\partial\boldsymbol{\delta}_i}\right)^{\mathrm{T}}\dot{\boldsymbol{\delta}}_i-\left(\frac{\partial V_i^k}{\partial\boldsymbol{\delta}_i}\right)^{\mathrm{T}}\dot{\boldsymbol{\delta}}_i\right)\mathrm{d}\tau.\end{aligned}\tag{27}$$

Based on Eq. (24), we have

$$\begin{aligned}&\left(\frac{\partial V_i^{k+1}}{\partial\boldsymbol{\delta}_i}\right)^{\mathrm{T}}\left(\boldsymbol{A}\boldsymbol{\delta}_i+(d_i+g_i)\boldsymbol{B}\boldsymbol{u}_i^{k+1}\right.\\&\left.-\sum_{j\in N_i}e_{ij}\boldsymbol{B}\boldsymbol{u}_j^{k+1}\right)+\boldsymbol{\delta}_i^{\mathrm{T}}\boldsymbol{Q}_{ii}\boldsymbol{\delta}_i\\&+R_i(\boldsymbol{u}_i^{k+1})+\sum_{j\in N_i}R_i(\boldsymbol{u}_j^{k+1})=0.\end{aligned}\tag{28}$$

Subtracting and adding Eqs. (24) and (28) to the right-hand side of Eq. (27), we have

$$\begin{aligned}&\left(\frac{\partial V_i^{k+1}}{\partial\boldsymbol{\delta}_i}\right)^{\mathrm{T}}\dot{\boldsymbol{\delta}}_i-\left(\frac{\partial V_i^k}{\partial\boldsymbol{\delta}_i}\right)^{\mathrm{T}}\dot{\boldsymbol{\delta}}_i\\=&\left(\frac{\partial V_i^{k+1}}{\partial\boldsymbol{\delta}_i}\right)^{\mathrm{T}}\sum_{j\in N_i}e_{ij}\boldsymbol{B}\left(\boldsymbol{u}_j^{k+1}-\boldsymbol{u}_j^k\right)-R_i(\boldsymbol{u}_i^{k+1})\\&-\sum_{j\in N_i}R_i(\boldsymbol{u}_j^{k+1})+R_i(\boldsymbol{u}_i^k)+\sum_{j\in N_i}R_i(\boldsymbol{u}_j^k)\\&-\left(\frac{\partial V_i^k}{\partial\boldsymbol{\delta}_i}\right)^{\mathrm{T}}(d_i+g_i)\boldsymbol{B}\left(\boldsymbol{u}_i^{k+1}-\boldsymbol{u}_i^k\right).\end{aligned}\tag{29}$$

For Eq. (29), we have

$$\begin{aligned}
&R_i(\boldsymbol{u}_i^k)-R_i(\boldsymbol{u}_i^{k+1})+\left(\frac{\partial V_i^k}{\partial \boldsymbol{\delta}_i}\right)^{\mathrm{T}}(d_i+g_i)\boldsymbol{B}\left(\boldsymbol{u}_i^k-\boldsymbol{u}_i^{k+1}\right)\\
=&R_i(\boldsymbol{u}_i^k)-R_i(\boldsymbol{u}_i^{k+1})\\
&-2\lambda_i\left(\Psi^{-1}\left(\frac{\boldsymbol{u}_i^{k+1}}{\lambda_i}\right)\right)^{\mathrm{T}}\boldsymbol{R}_{ii}\left(\boldsymbol{u}_i^k-\boldsymbol{u}_i^{k+1}\right)\\
=&2\lambda_i^2\int_{\boldsymbol{u}_i^{k+1}/\lambda_i}^{\boldsymbol{u}_i^k/\lambda_i}\left(\Psi^{-1}(\boldsymbol{s})\right)^{\mathrm{T}}\boldsymbol{R}_{ii}\mathrm{d}\boldsymbol{s}\\
&-2\lambda_i^2\int_{\boldsymbol{u}_i^{k+1}/\lambda_i}^{\boldsymbol{u}_i^k/\lambda_i}\left(\Psi^{-1}\left(\frac{\boldsymbol{u}_i^{k+1}}{\lambda_i}\right)\right)^{\mathrm{T}}\boldsymbol{R}_{ii}\mathrm{d}\boldsymbol{s}\\
=&2\lambda_i^2\int_{\boldsymbol{u}_i^{k+1}/\lambda_i}^{\boldsymbol{u}_i^k/\lambda_i}\left(\Psi^{-1}(\boldsymbol{s})-\Psi^{-1}\left(\frac{\boldsymbol{u}_i^{k+1}}{\lambda_i}\right)\right)^{\mathrm{T}}\boldsymbol{R}_{ii}\mathrm{d}\boldsymbol{s}
\end{aligned}\tag{30}$$

and

$$\begin{aligned}
&\sum_{j\in N_i}R_i(\boldsymbol{u}_j^k)-\sum_{j\in N_i}R_i(\boldsymbol{u}_j^{k+1})\\
&=\sum_{j\in N_i}2\lambda_j^2\int_{\boldsymbol{u}_j^{k+1}/\lambda_j}^{\boldsymbol{u}_j^k/\lambda_j}\left(\Psi^{-1}(\boldsymbol{s})\right)^{\mathrm{T}}\boldsymbol{R}_{ij}\mathrm{d}\boldsymbol{s}.
\end{aligned}\tag{31}$$

Therefore, we can rewrite Eq. (27) as

$$\begin{aligned}
&V_i^k-V_i^{k+1}\\
=&\int_t^{\infty}\left(\sum_{j\in N_i}2\lambda_j^2\int_{\boldsymbol{u}_j^{k+1}/\lambda_j}^{\boldsymbol{u}_j^k/\lambda_j}\left(\Psi^{-1}(\boldsymbol{s})\right)^{\mathrm{T}}\boldsymbol{R}_{ij}\mathrm{d}\boldsymbol{s}\right.\\
&+2\lambda_i^2\int_{\boldsymbol{u}_i^{k+1}/\lambda_i}^{\boldsymbol{u}_i^k/\lambda_i}\left(\Psi^{-1}(\boldsymbol{s})-\Psi^{-1}\left(\frac{\boldsymbol{u}_i^{k+1}}{\lambda_i}\right)\right)^{\mathrm{T}}\boldsymbol{R}_{ii}\mathrm{d}\boldsymbol{s}\\
&\left.+\left(\frac{\partial V_i^{k+1}}{\partial \boldsymbol{\delta}_i}\right)^{\mathrm{T}}\sum_{j\in N_i}e_{ij}\boldsymbol{B}\left(\boldsymbol{u}_j^{k+1}-\boldsymbol{u}_j^k\right)\right)\mathrm{d}\tau.
\end{aligned}\tag{32}$$

Based on Eq. (22) and inequality (23), it can be derived that

$$\int_{\boldsymbol{u}_i^{k+1}/\lambda_i}^{\boldsymbol{u}_i^k/\lambda_i}\left(\Psi^{-1}(\boldsymbol{s})-\Psi^{-1}\left(\frac{\boldsymbol{u}_i^{k+1}}{\lambda_i}\right)\right)^{\mathrm{T}}\boldsymbol{R}_{ii}\mathrm{d}\boldsymbol{s}>0.\tag{33}$$

For Eq. (32), if e_{ij} and $\sigma_{\max}(\boldsymbol{R}_{ij})$ are small enough, where $\sigma_{\max}(\cdot)$ represents the maximum singular value of the matrix, it can be derived that $V_i^k\geq V_i^{k+1}$. According to Eq. (9), we have $V_i^k\geq 0$. Therefore, V_i^k is convergent as $k\to\infty$. As V_i^* is the optimal iterative value function obtained by minimizing Eq. (9), we have $\lim\limits_{k\to\infty}V_i^k=V_i^{\infty}\geq V_i^*$. Therefore, it can be derived that V_i^k converges to V_i^*, where V_i^* is the solution to Eq. (12). The proof is completed.

However, system matrices are still required to solve Eqs. (24) and (25). A novel off-policy reinforcement learning method will be presented in the next subsection without the information of system matrices.

3.3 Model-free off-policy reinforcement learning method for solving the HJB equation

We can rewrite the tracking error dynamics, represented in Eq. (4), as follows:

$$\begin{aligned}
\dot{\boldsymbol{\delta}}_i=&\boldsymbol{A}\boldsymbol{\delta}_i+(d_i+g_i)\boldsymbol{B}\left(\boldsymbol{u}_i-\boldsymbol{u}_i^k\right)+(d_i+g_i)\boldsymbol{B}\boldsymbol{u}_i^k\\
&-\sum_{j\in N_i}e_{ij}\boldsymbol{B}\boldsymbol{u}_j^k-\sum_{j\in N_i}e_{ij}\boldsymbol{B}\left(\boldsymbol{u}_j-\boldsymbol{u}_j^k\right).
\end{aligned}\tag{34}$$

Taking the derivative of V_i^k along system (34), we have

$$\begin{aligned}
\dot{V}_i^k=&\left(\frac{\partial V_i^k}{\partial \boldsymbol{\delta}_i}\right)^{\mathrm{T}}\dot{\boldsymbol{\delta}}_i\\
=&\left(\frac{\partial V_i^k}{\partial \boldsymbol{\delta}_i}\right)^{\mathrm{T}}\left(\boldsymbol{A}\boldsymbol{\delta}_i-\sum_{j\in N_i}e_{ij}\boldsymbol{B}\boldsymbol{u}_j^k+(d_i+g_i)\boldsymbol{B}\boldsymbol{u}_i^k\right)\\
&+\left(\frac{\partial V_i^k}{\partial \boldsymbol{\delta}_i}\right)^{\mathrm{T}}(d_i+g_i)\boldsymbol{B}\left(\boldsymbol{u}_i-\boldsymbol{u}_i^k\right)\\
&-\left(\frac{\partial V_i^k}{\partial \boldsymbol{\delta}_i}\right)^{\mathrm{T}}\sum_{j\in N_i}e_{ij}\boldsymbol{B}\left(\boldsymbol{u}_j-\boldsymbol{u}_j^k\right).
\end{aligned}\tag{35}$$

According to Eq. (25), we have

$$\boldsymbol{\Delta}_i^{k+1}=\frac{1}{2\lambda_i}(d_i+g_i)\boldsymbol{R}_{ii}^{-1}\boldsymbol{B}^{\mathrm{T}}\frac{\partial V_i^k}{\partial \boldsymbol{\delta}_i}.\tag{36}$$

Then, substituting Eqs. (24) and (36) into Eq. (35), it can be derived that

$$\begin{aligned}
&V_i^k(\boldsymbol{\delta}_i(t'))-V_i^k(\boldsymbol{\delta}_i(t))\\
=&\int_t^{t'}\dot{V}_i^k\mathrm{d}\tau\\
=&-\int_t^{t'}\boldsymbol{\delta}_i^{\mathrm{T}}\boldsymbol{Q}_{ii}\boldsymbol{\delta}_i\mathrm{d}\tau-\int_t^{t'}R_i(\boldsymbol{u}_i^k)\mathrm{d}\tau-\int_t^{t'}\sum_{j\in N_i}R_i(\boldsymbol{u}_j^k)\mathrm{d}\tau\\
&+\int_t^{t'}2\lambda_i\left(\boldsymbol{\Delta}_i^{k+1}\right)^{\mathrm{T}}\boldsymbol{R}_{ii}(\boldsymbol{u}_i-\boldsymbol{u}_i^k)\mathrm{d}\tau\\
&-\int_t^{t'}\sum_{j\in N_i}e_{ij}\frac{2\lambda_i}{d_i+g_i}\left(\boldsymbol{\Delta}_i^{k+1}\right)^{\mathrm{T}}\boldsymbol{R}_{ii}(\boldsymbol{u}_j-\boldsymbol{u}_j^k)\mathrm{d}\tau.
\end{aligned}\tag{37}$$

Therefore, it can be seen that the system matrices are not included in Eq. (37). Based on the Weierstrass high-order approximation theorem (Abu-Khalaf and Lewis, 2005), the critic and actor NNs can be introduced as

$$\hat{V}_i^k = (\phi_i(\boldsymbol{\delta}_i))^{\mathrm{T}} \boldsymbol{W}_{vi}^k, \tag{38}$$

$$\hat{\Delta}_{il_1}^k = (\varphi_{uil_1}(\boldsymbol{\delta}_i))^{\mathrm{T}} \boldsymbol{W}_{uil_1}^k, \tag{39}$$

where $\phi_i \in \mathbb{R}^{h_v}$ and $\varphi_{uil_1} \in \mathbb{R}^{h_{ul_1}}$ $(l_1 = 1, 2, \ldots, m)$ are activation functions. $\boldsymbol{W}_{vi}^k$ and $\boldsymbol{W}_{uil_1}^k$ are constant weights. Eq. (39) can be written in the following compact form:

$$\begin{aligned}\hat{\boldsymbol{\Delta}}_i^k &= \begin{bmatrix} \hat{\Delta}_{i1}^k & \hat{\Delta}_{i2}^k & \ldots & \hat{\Delta}_{im}^k \end{bmatrix}^{\mathrm{T}} \\ &= \begin{bmatrix} (\varphi_{ui1}(\boldsymbol{\delta}_i))^{\mathrm{T}} \boldsymbol{W}_{ui1}^k & (\varphi_{ui2}(\boldsymbol{\delta}_i))^{\mathrm{T}} \boldsymbol{W}_{ui2}^k & \ldots \\ (\varphi_{uim}(\boldsymbol{\delta}_i))^{\mathrm{T}} \boldsymbol{W}_{uim}^k \end{bmatrix}^{\mathrm{T}}.\end{aligned} \tag{40}$$

Substituting Eqs. (38) and (39) into Eq. (37), we can obtain Eq. (41) (on the top of the next page), where $\hat{\boldsymbol{u}}_i^k = -\lambda_i \Psi(\hat{\boldsymbol{\Delta}}_i^k)$, σ_i^k represents the residual error, and $\boldsymbol{\delta}_i' = \boldsymbol{\delta}_i(t')$. Then, Eq. (41) can be written in a simplified form as follows:

$$\sigma_i^k = \boldsymbol{\rho}_{i,[t,t']}^k \boldsymbol{W}_i^{k+1} - \pi_{i,[t,t']}^k, \tag{42}$$

where

$$\begin{aligned}\pi_{i,[t,t']}^k = &\int_t^{t'} \boldsymbol{\delta}_i^{\mathrm{T}} \boldsymbol{Q}_{ii} \boldsymbol{\delta}_i \mathrm{d}\tau + \int_t^{t'} R_i(\hat{\boldsymbol{u}}_i^k) \mathrm{d}\tau \\ &+ \int_t^{t'} \sum_{j \in N_i} R_i(\hat{\boldsymbol{u}}_j^k) \mathrm{d}\tau,\end{aligned}$$

$$\boldsymbol{W}_i^{k+1} = \begin{bmatrix} (\boldsymbol{W}_{vi}^k)^{\mathrm{T}} & (\boldsymbol{W}_{ui1}^{k+1})^{\mathrm{T}} & \ldots & (\boldsymbol{W}_{uim}^{k+1})^{\mathrm{T}} \end{bmatrix}^{\mathrm{T}},$$

$$\boldsymbol{\rho}_{i,[t,t']}^k = \begin{bmatrix} \boldsymbol{\rho}_{vi}^k & \boldsymbol{\rho}_{ui1}^k & \ldots & \boldsymbol{\rho}_{uim}^k \end{bmatrix},$$

with

$$\boldsymbol{\rho}_{vi}^k = (\phi_i(\boldsymbol{\delta}_i) - \phi_i(\boldsymbol{\delta}_i'))^{\mathrm{T}},$$

$$\begin{aligned}\boldsymbol{\rho}_{uil_2}^k = &2\lambda_i \sum_{l_1=1}^{m} r_{ii,l_1 l_2} \\ &\cdot \int_t^{t'} \left(u_{il_1} + \lambda_i \Psi\left(\hat{\Delta}_{il_1}^k\right)\right) (\varphi_{uil_2}(\boldsymbol{\delta}_i))^{\mathrm{T}} \mathrm{d}\tau \\ &- \frac{2\lambda_i}{d_i + g_i} \sum_{l_1=1}^{m} r_{ii,l_1 l_2} \\ &\cdot \int_t^{t'} \sum_{j \in N_i} e_{ij} \left(u_{jl_1} + \lambda_j \Psi\left(\hat{\Delta}_{jl_1}^k\right)\right) (\varphi_{uil_2}(\boldsymbol{\delta}_i))^{\mathrm{T}} \mathrm{d}\tau,\end{aligned}$$

for $l_2 = 1, 2, \ldots, m$.

We would like to determine matrix $\boldsymbol{W}_i^{k+1}$ that can minimize the residual error σ_i^k. For a positive integer υ, we can define $\boldsymbol{\Gamma}_i^k$ and $\boldsymbol{\Pi}_i^k$ as

$$\boldsymbol{\Gamma}_i^k = \left[\left(\boldsymbol{\rho}_{i,[t_0,t_1]}^k\right)^{\mathrm{T}} \left(\boldsymbol{\rho}_{i,[t_1,t_2]}^k\right)^{\mathrm{T}} \cdots \left(\boldsymbol{\rho}_{i,[t_{\upsilon-1},t_\upsilon]}^k\right)^{\mathrm{T}}\right]^{\mathrm{T}},$$

$$\boldsymbol{\Pi}_i^k = \begin{bmatrix} \pi_{i,[t_0,t_1]}^k & \pi_{i,[t_1,t_2]}^k & \cdots & \pi_{i,[t_{\upsilon-1},t_\upsilon]}^k \end{bmatrix}^{\mathrm{T}}.$$

Based on the least square approach, it can be obtained that

$$\boldsymbol{W}_i^{k+1} = \left((\boldsymbol{\Gamma}_i^k)^{\mathrm{T}} \boldsymbol{\Gamma}_i^k\right)^{-1} (\boldsymbol{\Gamma}_i^k)^{\mathrm{T}} \boldsymbol{\Pi}_i^k. \tag{43}$$

A model-free off-policy reinforcement learning algorithm can be provided in Algorithm 2.

Algorithm 2 Model-free off-policy reinforcement learning

Initialization: Choose the initial stabilizing network weights $\boldsymbol{W}_{uil_1}^0$ for $i = 1, 2, \ldots, N$, $l_1 = 1, 2, \ldots, m$; choose the computation precision ε

Iteration:

1: Employ the control laws $\boldsymbol{u}_i$ $(i = 1, 2, \ldots, N)$ with exploration noises to system (34) on the time interval $[t_0, t_\upsilon]$. Collect the system data $\{\boldsymbol{\delta}_i, \boldsymbol{u}_i\}$ for $i = 1, 2, \ldots, N$

2: Let iteration index $k = 0$

3: Compute $\boldsymbol{\rho}_{i,[t_0,t_1]}^k$, $\boldsymbol{\rho}_{i,[t_1,t_2]}^k$, …, $\boldsymbol{\rho}_{i,[t_{\upsilon-1},t_\upsilon]}^k$ and $\pi_{i,[t_0,t_1]}^k$, $\pi_{i,[t_1,t_2]}^k$, …, $\pi_{i,[t_{\upsilon-1},t_\upsilon]}^k$

4: Compute $\boldsymbol{W}_i^{k+1}$ according to Eq. (43)

5: Let iteration index $k = k + 1$; repeat steps 2 and 3 until $\left\|\boldsymbol{W}_i^{k+1} - \boldsymbol{W}_i^k\right\| \le \varepsilon$ for $k \ge 1$

6: **return** $\boldsymbol{W}_i^k, i = 1, 2, \ldots, N$

Lemma 1 For system (4), if the iterative value functions and iterative control laws are designed as Eqs. (38) and (39), respectively, where the weights $\boldsymbol{W}_{vi}^k$ and $\boldsymbol{W}_{uil_1}^k$ $(l_1 = 1, 2, \ldots, m)$ are updated as Algorithm 2, $\lim\limits_{k \to \infty} \hat{\boldsymbol{u}}_i^k = \boldsymbol{u}_i^*$ $(i = 1, 2, \ldots, N)$.

The proof process can be found in Li JN et al. (2017) and Qin et al. (2019), and thus is omitted here.

Remark 3 In Algorithm 2, the selection of the control laws $\boldsymbol{u}_i$ $(i = 1, 2, \ldots, N)$ is the key to the convergence of the algorithm. Generally, the control laws are selected as $\boldsymbol{u}_i = -\boldsymbol{K}_i \boldsymbol{\delta}_i + \boldsymbol{\xi}_i$ $(i = 1, 2, \ldots, N)$, where $\boldsymbol{\xi}_i$ is the exploration noise and $\boldsymbol{K}_i$ is the stabilizing gain matrix.

Remark 4 For traditional on-policy integral reinforcement learning methods (Vrabie and Lewis, 2011;

$$\sigma_i^k = (\phi_i(\boldsymbol{\delta}_i) - \phi_i(\boldsymbol{\delta}_i'))^{\mathrm{T}} \boldsymbol{W}_{vi}^k - \int_t^{t'} \boldsymbol{\delta}_i^{\mathrm{T}} \boldsymbol{Q}_{ii} \boldsymbol{\delta}_i \mathrm{d}\tau - \int_t^{t'} R_i(\hat{\boldsymbol{u}}_i^k)\mathrm{d}\tau - \int_t^{t'} \sum_{j\in N_i} R_i(\hat{\boldsymbol{u}}_j^k)\mathrm{d}\tau$$
$$+ 2\lambda_i \sum_{l_1=1}^{m}\sum_{l_2=1}^{m} r_{ii,l_1l_2} \int_t^{t'} \left(u_{il_1} + \lambda_i \Psi\left(\hat{\Delta}_{il_1}^k\right)\right) (\varphi_{uil_2}(\boldsymbol{\delta}_i))^{\mathrm{T}} \boldsymbol{W}_{uil_2}^{k+1} \mathrm{d}\tau$$
$$- \frac{2\lambda_i}{d_i + g_i} \sum_{l_1=1}^{m}\sum_{l_2=1}^{m} r_{ii,l_1l_2} \int_t^{t'} \sum_{j\in N_i} e_{ij} \left(u_{jl_1} + \lambda_j \Psi\left(\hat{\Delta}_{jl_1}^k\right)\right) (\varphi_{uil_2}(\boldsymbol{\delta}_i))^{\mathrm{T}} \boldsymbol{W}_{uil_2}^{k+1} \mathrm{d}\tau. \quad (41)$$

Liu et al., 2021), the performance index function is evaluated using the inaccurate data, which cause a biased estimation. The presented off-policy reinforcement learning method can avoid this problem and thus obtain results with higher accuracy.

4 Numerical analysis

In this section, a simulation example is provided to show the good performance of the presented method. The structure of the multi-agent systems is shown in Fig. 1, with the following dynamics:

$$\boldsymbol{A} = \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix}, \ \boldsymbol{B} = \begin{bmatrix} 1 \\ 1 \end{bmatrix}.$$

The Laplacian matrix and the pinning matrix are given as

$$\boldsymbol{L} = \begin{bmatrix} 1 & -1 & 0 \\ 0 & 1 & -1 \\ -1 & -1 & 2 \end{bmatrix}, \ \boldsymbol{G} = \begin{bmatrix} 1 & & \\ & 0 & \\ & & 0 \end{bmatrix}.$$

We define the weight matrices of the performance index function, represented by Eq. (5), as follows:

$$\boldsymbol{Q}_{11} = \boldsymbol{I}_2, \ \boldsymbol{Q}_{22} = 2\boldsymbol{I}_2, \ \boldsymbol{Q}_{33} = 1.5\boldsymbol{I}_2,$$
$$R_{11} = 2, \ R_{22} = 1, \ R_{33} = 3,$$
$$R_{12} = 0.02, \ R_{23} = 0.01, \ R_{31} = 0.03, \ R_{32} = 0.01.$$

The simulation is performed with $\boldsymbol{x}_0(0) = [1 \quad 1]^{\mathrm{T}}$, $\boldsymbol{x}_1(0) = [0.5 \quad -0.5]^{\mathrm{T}}$, $\boldsymbol{x}_2(0) = [1 \quad -0.5]^{\mathrm{T}}$, $\boldsymbol{x}_3(0) = [2 \quad -1]^{\mathrm{T}}$, $\lambda_1 = 2$, $\lambda_2 = 1.5$, $\lambda_3 = 3$. First, we collect the system data $\{\boldsymbol{\delta}_i, \boldsymbol{u}_i\}$ every 0.01 s for $i = 1, 2, 3$. Then, we solve Eq. (43) iteratively based on the collected system data. The activation functions $\phi_i(\boldsymbol{\delta}_i)$ and $\varphi_{ui}(\boldsymbol{\delta}_i)$ are chosen as

$$\phi_i(\boldsymbol{\delta}_i) = [\delta_{i1}^2 \quad \delta_{i1}\delta_{i2} \quad \delta_{i2}^2]^{\mathrm{T}},$$
$$\varphi_{ui}(\boldsymbol{\delta}_i) = [\delta_{i1} \quad \delta_{i2} \quad \delta_{i1}^2 \quad \delta_{i1}\delta_{i2} \quad \delta_{i2}^2]^{\mathrm{T}}.$$

The simulation results are shown in Figs. 2–6. The weights of critic and actor NNs are shown in Figs. 2 and 3, respectively, which show the stability of Algorithm 2. The synchronization error curves are provided in Fig. 4, and the three-dimensional curves are provided in Fig. 5. From Figs. 4 and 5, it can be derived that the optimal synchronization control is achieved. The control curves are shown in Fig. 6, verifying that the control constraint is satisfied.

5 Conclusions

The nonzero-sum game problem of multi-agent systems with input saturation has been studied

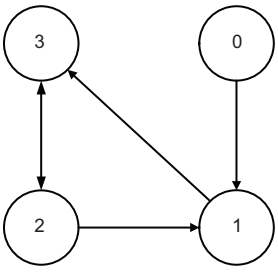

Fig. 1 Structure of the multi-agent systems

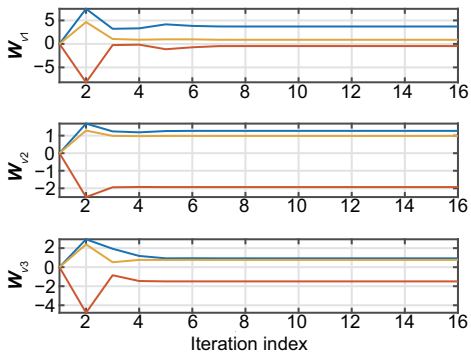

Fig. 2 Weights of critic neural networks of the multi-agent systems

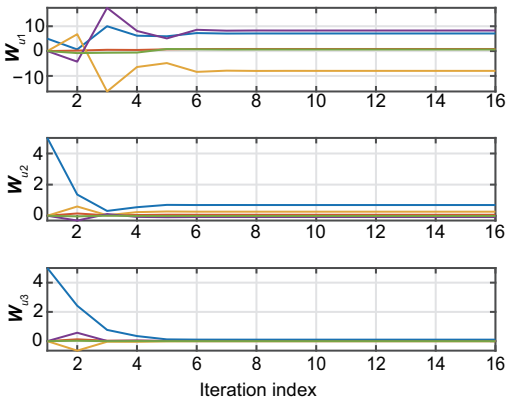

Fig. 3 Weights of actor neural networks of the multi-agent systems

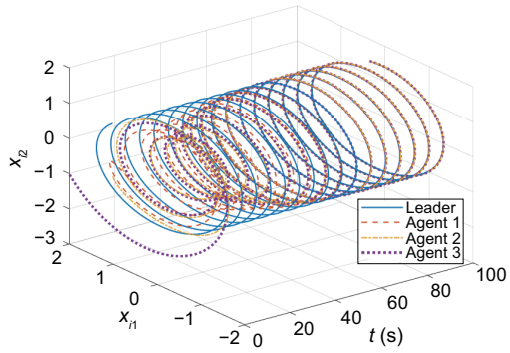

Fig. 5 Three-dimensional curves of the multi-agent systems

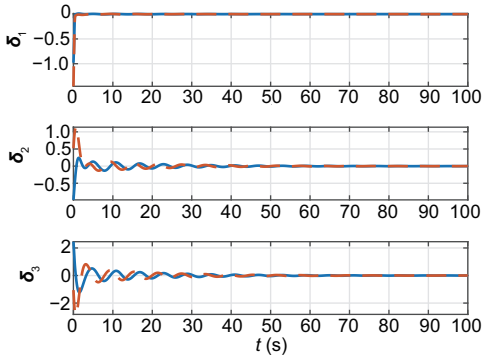

Fig. 4 Synchronization errors of the multi-agent systems

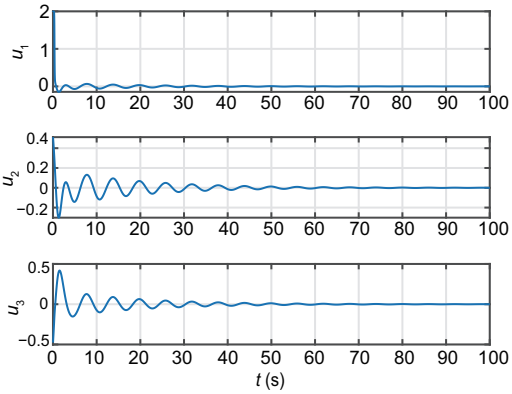

Fig. 6 Control laws of the multi-agent systems

based on the model-free off-policy reinforcement learning method. It is shown that the presented off-policy reinforcement learning algorithm can make the iterative control laws converge to the Nash equilibrium without the information of system models. The simulation results showed the good performance of the presented method.

Contributors

Hongyang LI designed the method, conducted the simulation, and drafted the paper. Qinglai WEI revised and finalized the paper.

Compliance with ethics guidelines

Hongyang LI and Qinglai WEI declare that they have no conflict of interest.

References

Abu-Khalaf M, Lewis FL, 2005. Nearly optimal control laws for nonlinear systems with saturating actuators using a neural network HJB approach. *Automatica*, 41(5):779-791. https://doi.org/10.1016/j.automatica.2004.11.034

Bertsekas DP, 2007. Dynamic Programming and Optimal Control. Athena Scientific, Belmont, USA.

Cao MT, Xiao F, Wang L, 2015. Event-based second-order consensus control for multi-agent systems via synchronous periodic event detection. *IEEE Trans Autom Contr*, 60(9):2452-2457. https://doi.org/10.1109/TAC.2015.2390553

Du HB, He YG, Cheng YY, 2014. Finite-time synchronization of a class of second-order nonlinear multi-agent systems using output feedback control. *IEEE Trans Circ Syst I*, 61(6):1778-1788. https://doi.org/10.1109/TCSI.2013.2295012

Garcia E, Cao YC, Casbeer D, 2017. Periodic event-triggered synchronization of linear multi-agent systems with communication delays. *IEEE Trans Autom Contr*, 62(1):366-371. https://doi.org/10.1109/TAC.2016.2555484

Han YJ, Lu WL, Chen TP, 2013. Cluster consensus in discrete-time networks of multi-agents with inter-cluster nonidentical inputs. *IEEE Trans Neur Netw Learn Syst*, 24(4):566-578. https://doi.org/10.1109/TNNLS.2013.2237786

He WL, Gao XY, Zhong WM, et al., 2018. Secure impulsive synchronization control of multi-agent systems under deception attacks. *Inform Sci*, 459:354-368. https://doi.org/10.1016/j.ins.2018.04.020

Jiao Q, Modares H, Xu SY, et al., 2016. Multi-agent zero-sum differential graphical games for disturbance rejection in distributed control. *Automatica*, 69:24-34. https://doi.org/10.1016/j.automatica.2016.02.002

Li JN, Modares H, Chai TY, et al., 2017. Off-policy reinforcement learning for synchronization in multiagent graphical games. *IEEE Trans Neur Netw Learn Syst*, 28(10):2434-2445. https://doi.org/10.1109/TNNLS.2016.2609500

Li JQ, Wang QL, Su YX, et al., 2021. Robust distributed model predictive consensus of discrete-time multi-agent systems: a self-triggered approach. *Front Inform Technol Electron Eng*, 22(8):1068-1079. https://doi.org/10.1631/FITEE.2000182

Liu DR, Xue S, Zhao B, et al., 2021. Adaptive dynamic programming for control: a survey and recent advances. *IEEE Trans Syst Man Cybern Syst*, 51(1):142-160. https://doi.org/ 10.1109/TSMC.2020.3042876

Ma HJ, Yang GH, 2016. Adaptive fault tolerant control of cooperative heterogeneous systems with actuator faults and unreliable interconnections. *IEEE Trans Autom Contr*, 61(11):3240-3255. https://doi.org/10.1109/TAC.2015.2507864

Qin JH, Li M, Shi Y, et al., 2019. Optimal synchronization control of multiagent systems with input saturation via off-policy reinforcement learning. *IEEE Trans Neur Netw Learn Syst*, 30(1):85-96. https://doi.org/10.1109/TNNLS.2018.2832025

Rehák B, Lynnyk V, 2021. Leader-following synchronization of a multi-agent system with heterogeneous delays. *Front Inform Technol Electron Eng*, 22(1):97-106. https://doi.org/10.1631/FITEE.2000207

Thunberg J, Song W, Monitijano E, et al., 2014. Distributed attitude synchronization control of multi-agent systems with switching topologies. *Automatica*, 50(3):832-840. https://doi.org/10.1016/j.automatica.2014.02.002

Vamvoudakis KG, Lewis FL, Hudas GR, 2012. Multi-agent differential graphical games: online adaptive learning solution for synchronization with optimality. *Automatica*, 48(8):1598-1611. https://doi.org/10.1016/j.automatica.2012.05.074

Vrabie D, Lewis F, 2011. Adaptive dynamic programming for online solution of a zero-sum differential game. *J Contr Theory Appl*, 9(3):353-360. https://doi.org/10.1007/s11768-011-0166-4

Wang FY, Zhang HG, Liu DR, 2009. Adaptive dynamic programming: an introduction. *IEEE Comput Intell Mag*, 4(2):39-47. https://doi.org/10.1109/MCI.2009.932261

Wei QL, Liu DR, 2014. Adaptive dynamic programming for optimal tracking control of unknown nonlinear systems with application to coal gasification. *IEEE Trans Autom Sci Eng*, 11(4):1020-1036. https://doi.org/10.1109/TASE.2013.2284545

Wei QL, Wang FY, Liu DR, et al., 2014. Finite-approximation-error-based discrete-time iterative adaptive dynamic programming. *IEEE Trans Cybern*, 44(12):2820-2833. https://doi.org/10.1109/TCYB.2014.2354377

Wei QL, Liu DR, Lewis FL, 2015. Optimal distributed synchronization control for continuous-time heterogeneous multi-agent differential graphical games. *Inform Sci*, 317:96-113. https://doi.org/10.1016/j.ins.2015.04.044

Wei QL, Liu DR, Lin HQ, 2016. Value iteration adaptive dynamic programming for optimal control of discrete-time nonlinear systems. *IEEE Trans Cybern*, 46(3):840-853. https://doi.org/10.1109/TCYB.2015.2492242

Wei QL, Lewis FL, Sun QY, et al., 2017. Discrete-time deterministic Q-learning: a novel convergence analysis. *IEEE Trans Cybern*, 47(5):1224-1237. https://doi.org/10.1109/TCYB.2016.2542923

Wei QL, Lewis FL, Liu DR, et al., 2018. Discrete-time local value iteration adaptive dynamic programming: convergence analysis. *IEEE Trans Syst Man Cybern Syst*, 48(6):875-891. https://doi.org/10.1109/TSMC.2016.2623766

Wei QL, Li HY, Wang FY, 2020. Parallel control for continuous-time linear systems: a case study. *IEEE/CAA J Autom Sin*, 7(4):919-928. https://doi.org/10.1109/JAS.2020.1003216

Wei QL, Wang X, Zhong XN, et al., 2021. Consensus control of leader-following multi-agent systems in directed topology with heterogeneous disturbances. *IEEE/CAA J Autom Sin*, 8(2):423-431. https://doi.org/10.1109/JAS.2021.1003838

Wieland P, Sepulchre R, Allgöwer F, 2011. An internal model principle is necessary and sufficient for linear output synchronization. *Automatica*, 47(5):1068-1074. https://doi.org/10.1016/j.automatica.2011.01.081

Yang JY, Xi F, Ma J, 2019. Model-based edge-event-triggered containment control under directed topologies. *IEEE Trans Cybern*, 49(7):2556-2567. https://doi.org/10.1109/TCYB.2018.2828645

Yang N, Xiao JW, Xiao L, et al., 2019. Non-zero sum differential graphical game: cluster synchronisation for multi-agents with partially unknown dynamics. *Int J Contr*, 92(10):2408-2419. https://doi.org/10.1080/00207179.2018.1441550

Zhang HG, Zhang JL, Yang GH, et al., 2015. Leader-based optimal coordination control for the consensus problem of multiagent differential games via fuzzy adaptive dynamic programming. *IEEE Trans Fuzzy Syst*, 23(1):152-163. https://doi.org/10.1109/TFUZZ.2014.2310238

Zhang KQ, Yang ZR, Başar T, 2021. Decentralized multi-agent reinforcement learning with networked agents: recent advances. *Front Inform Technol Electron Eng*, 22(6):802-814. https://doi.org/10.1631/FITEE.1900661

Zhang LD, Wang B, Liu ZX, et al., 2019. Motion planning of a quadrotor robot game using a simulation-based projected policy iteration method. *Front Inform Technol Electron Eng*, 20(4):525-537. https://doi.org/10.1631/FITEE.1800571

Zhao DY, Zhu QM, Li N, et al., 2014. Synchronized control with neuro-agents for leader-follower based multiple robotic manipulators. *Neurocomputing*, 124:149-161. https://doi.org/10.1016/j.neucom.2013.07.016