Applied Mathematics and Computation 369 (2020) 124821

Contents lists available at ScienceDirect

Applied Mathematics and Computation

journal homepage: www.elsevier.com/locate/amc

A novel optimal bipartite consensus control scheme for unknown multi-agent systems via model-free reinforcement learning

Zhinan Peng[a], Jiangping Hu[a], Kaibo Shi[b,*], Rui Luo[a], Rui Huang[a], Bijoy Kumar Ghosh[a,c], Jiuke Huang[d]

[a] School of Automation Engineering, University of Electronic Science and Technology of China, Chengdu 611731, China
[b] School of Information Science and Engineering, Chengdu University, Chengdu, 610106, China
[c] Department of Mathematics and Statistics, Texas Tech University, Lubbock, TX 79409-1042, USA
[d] School of Engineering, Vanderbilt University, Nashville, TN 37235-1826, USA

ARTICLE INFO

Article history:
Received 26 March 2019
Revised 12 August 2019
Accepted 6 October 2019
Available online 6 November 2019

Keywords:
Optimal bipartite consensus control
Multi-agent systems
Coopetition network
Model-free
Reinforcement learning

ABSTRACT

In this paper, the optimal bipartite consensus control (OBCC) problem is investigated for unknown multi-agent systems (MASs) with coopetition networks. A novel distributed OBCC scheme is proposed based on model-free reinforcement learning method to achieve OBCC, where the agent's dynamics are no longer required. First, The coopetition networks are applied to establish the cooperative and competitive interactions among agents, and then the OBCC problem is formulated by introducing local neighbor bipartite consensus errors and performance index functions (PIFs) for each agent. Second, in order to obtain the OBCC laws, a policy iteration algorithm (PIA) is employed to learn the solutions to discrete-time (DT) Hamilton-Jacobi-Bellman (HJB) equations. Third, to implement the proposed methods, we adopt a data-driven actor-critic-based neural networks (NNs) framework to approximate the control laws and the PIFs, respectively, in an online learning manner. Finally, some simulation results are given to demonstrate the effectiveness of the developed approaches.

1. Introduction

Recently, the cooperative control problem of MASs has received a surge of attention due to its wildly applications in UAV [1], smart grids and power systems [2], formation control in robotic systems [3,4] and so on [5–7]. For cooperative MASs, consensus problem is a very important topics [8–12], and some related researches have been investigated for various situations, such as leader-follower (LF) tracking control [13,14], fault-tolerant consensus [15,16], and so on. However, most of the researches on the MASs control problem always has a common assumption that the interaction among agents is cooperative. In contrast with cooperation, competition is the other inherent phenomenon. The cooperation and competition (coopetition for simplicity) coexist in social systems. For example, the confrontational situation is common in a two alliances (political parties) such that the opposing opinions are held by the two parties, where members of each party reach an agreement [17,18].

* Corresponding author.
E-mail address: skbs111@163.com (K. Shi).

https://doi.org/10.1016/j.amc.2019.124821
0096-3003/

Hence, the other type of consensus control problem, namely, bipartite consensus (BC) control, for coopetitive MASs has been intensively studied from various perspectives in recent years [19–22]. For instance, some adaptive control based methods were utilized for BC control of reaction-diffusion neural networks (NNs) in [23] and high-order MASs in [24]. The authors in [25] also studied the corresponding BC problem under directed signed communication network. However, it is noted that the aforementioned results on BC controller designs of MASs strictly depends on system dynamics (SDs). Therefore, they usually need an important assumption that the knowledge of the SDs are known in advance. In fact, it has to take the real world cases into considerations, the information of accurate dynamics is always difficult to establish or obtain under the complex external environment. Meanwhile, in the aforementioned BC control of the MASs, the energy consumption optimization of each agent is not considered. From the above discussion and analysis, how to utilize a model-free (data-driven) based method to design control law and stabilize the unknown (i.e., black box) systems while minimizing the energy consumption, which is the first motivation of this paper.

Recently, intelligent technologies (ITs), such as machine learning, have been widely used in practical production and life, such as medical applications, flood debris forecasting and management, wireless sensor networks, and so on [26–28]. Especially, reinforcement learning (RL) as one of ITs, which has ability to tackle the optimal control (decision) problem for control systems in the case of energy consumption and cost. Adaptive dynamic programming (ADP) integrates adaptive control theory, optimization theory, which is regarded as the advanced tools of RL to handle optimal control problems [29–32]. Till now, there are two mainstream methods in this research field, one is value iteration algorithm (VIA), the other one is policy iteration algorithm (PIA), which are often used to approximate the optimal solution of the HJB equation indirectly in an iterative fashion [33]. For example, the authors in [34] studied the general nonlinear systems with the aid of VIA. The authors in [35,36] addressed the optimal control issue for DT nonlinear systems on account of VIA. In the meantime, the authors in [37] discussed the same theme for the continuous-time (CT) nonlinear systems based on PIA.

In the meantime, the ADP/RL-based MASs optimal control problems has attracted increasing attention for many researchers. Notice that some widespread interests have focused on several works on DT case [38–41]. Very interestingly, to handle with the situation where the SDs are unknown, model-free-RL/data-based methods were developed for various cooperative optimal control problems of MASs, such as consensus tracking control [42,43], synchronization in MASs graphical games [44], optimal containment control [45,46]. However, it is important to note that the above mentioned results are only focusing on cooperative relationship between agents. In fact, the coexistence of the competition and cooperation among agents is a universal phenomenon and an inevitable case. To the best of our knowledge, the OBCC problem for MASs with coopetition interactions has not been considered by using data-driven or model-free-RL based methods, which forms the second motivation of this paper.

From the above observations and analysis, this paper proposes a model-free-RL based controller designs to solve the OBCC problems for unknown DT LF MASs. Meanwhile, the general BC problem is transformed into the problem of OBCC by introducing PIFs, which relies on the local neighbour BC errors and the distributed control laws. The main contributions of the paper are summarized as follows: (1) A novel distributed OBCC scheme, to the best of our knowledge, is the first time to be proposed for unknown DT LF MASs; (2) A model-free-RL method based on the PIA is developed to obtain the optimal control laws, without requiring the accurate model of SDs; (3) An online learning mechanism, that is, actor-critic NNs, is established to estimate the PIFs and the control laws only using measurement system data instead of the accurate model of SDs.

2. Preliminaries

In this section, the basic signed graph theory for molding the MASs with coopetitive interactions is introduced, and then the OBCC problem is formulated.

2.1. Signed graph theory

We consider a coopetition communication network consisting of N agents, and define a signed graph (SG) as $\mathcal{F} = (\mathcal{V}, \mathcal{E}, \boldsymbol{A}_{\mathcal{F}})$, where the nonempty finite set of vertex is denoted as $\mathcal{V} = \{v_1, v_2, \ldots, v_N\}$, and $\mathcal{E} = \{(v_i, v_j)|v_i, v_j \in \mathcal{V}\} \subseteq \mathcal{V} \times \mathcal{V}$ denotes the nonempty finite set of arcs. Let $\boldsymbol{A}_{\mathcal{F}} = [a_{ij}] \in \mathcal{R}^{N\times N}$ be a weighted adjacency matrix with $-1, 0, 1$ elements. Arc $\mathcal{E}(i, j)$ is regarded as a communication flow from node i to node j, where a_{ij} is nonzero, that is, $a_{ij} \neq 0$ is equivalent to $(v_j, v_i) \subseteq \mathcal{V} \times \mathcal{V}$, which represents agent i is able to obtain agent j's information. If $a_{ij} > 0$, the interactions relationship between vertex i and j is *cooperative;* $a_{ij} < 0$ indicates the interactions between vertex i and vertex j is *competitive*; otherwise, $a_{ij} = 0$. Then, let $\mathcal{N}(i) = \{j|j \neq i, (v_j, v_i) \in \mathcal{E}\}$ be the set of node i's neighbours. The degree matrix $D = diag\{d_i\}$ of the SG $\boldsymbol{A}_{\mathcal{F}}$ is a diagonal matrix with $d_i = \sum_{j\in\mathcal{N}(i)} |a_{ij}|$. Thus, the Laplacian matrix of $\mathcal{F}$ can be calculated by $\mathcal{L} = -\boldsymbol{A}_{\mathcal{F}} + D \in \mathcal{R}^{N\times N}$.

In order to describe the relations between N agents and the leader, an augmented graph, i.e., $\tilde{\mathcal{F}} = (\tilde{\mathcal{V}}, \tilde{\mathcal{E}})$ is introduced where $\tilde{\mathcal{V}} = \{v_0, v_1, v_2, \ldots, v_N\}$ and $\tilde{\mathcal{E}} \subseteq \tilde{\mathcal{V}} \times \tilde{\mathcal{V}}$. Define a diagonal matrix $\mathcal{B}$, that is, $\mathcal{B} = diag\{b_1, \ldots, b_N\} \in \mathcal{R}^{N\times N}$. If $b_i > 0$, the agent i can obtain information form the leader. A directed path from node v_i to node v_j is denoted as a sequence of edges $\{(v_i, v_{k1}), (v_{k1}, v_{k2}), \ldots, (v_{km}, v_j)\}$. If the coopetition network $\mathcal{F}$ contains a spanning tree, then there is a root node such that $\mathcal{F}$ exists a directed path from the root to any other nodes.

The coopetition network $\mathcal{F}$ is called as structurally balanced (SB) [19], if the whole nodes in $\mathcal{V}$ can be able to be divided into two disjoint subsets, that is, $\mathcal{V}_1$, $\mathcal{V}_2$. They satisfy the following three conditions: 1) $\mathcal{V} = \mathcal{V}_1 \cup \mathcal{V}_2$, and $\mathcal{V} = \mathcal{V}_1 \cap \mathcal{V}_2 = \emptyset$, 2) If $\forall i, j \in \mathcal{V}_p$ $(p \in \{1, 2\})$, $a_{ij} \geq 0$, 3) If $\forall i \in \mathcal{V}_p$, $j \in \mathcal{V}_q$, $p \neq q$ $(p \in \{1, 2\})$, $a_{ij} \leq 0$.

2.2. Problem formulation

This paper considers a class of DT MASs consisting of N agents, whose dynamics are expressed by

$$x_i(k+1) = Ax_i(k) + B_i u_i(k), \tag{1}$$

where $x_i(k) \in \mathcal{R}^n$, $u_i(k) \in \mathcal{R}^{m_i}$, $(i = 1, 2, \ldots, N)$ denote the state variable of agent i and its control law, respectively. $A \in \mathcal{R}^n$ is the state matrix, $B_i \in \mathcal{R}^{n \times m_i}$ denotes the control input matrix. Herein, we assume that the system matrices A and B_i are considered as unknown matrices in the paper.

The state of leader (reference signal) is defined as $x_0(k)$ with the following dynamics

$$x_0(k+1) = Ax_0(k). \tag{2}$$

Assumption 1. The coopetition network $\mathcal{F}$ is SB and $\tilde{\mathcal{F}}$ has a spanning tree with the root note v_0 (leader).

Throughout this paper, we introduce the following lemma and definitions, which will be employed to introduce our main control problem.

Lemma 1 [47]**.** *According to the Assumption 1, it can be obtained that $\mathcal{L} + \mathcal{B}$ is a positive definite matrix.*

Definition 1 (BC control problem)**.** For each agent i, the goal of the BC control problem aims at designing the control law $u_i(k)$ only utilizing agent i's information and its neighbours', then the following conditions can be satisfied for $\forall i$, that is

$$\lim_{k \to \infty} (x_i(k) - x_0(k)) = 0, \tag{3}$$

for agent $i \in \mathcal{V}_1$,

$$\lim_{k \to \infty} (x_i(k) + x_0(k)) = 0, \tag{4}$$

for agent $i \in \mathcal{V}_2$.

For convenience of analysis, a gauge transformation matrix is defined by $S = diag\{s_1, s_2, \ldots, s_N\} \in \mathcal{R}^{N \times N}$, where $s_i = 1$ for $i \in \mathcal{V}_1$ and $s_i = -1$ for $i \in \mathcal{V}_2$.

Remark 1. In fact, the above Eqs. (3) and (4) can be rewrote as one equation format, that is

$$\lim_{k \to \infty} (x_i(k) - s_i x_0(k)) = 0,$$

where $s_i = 1$ for $i \in \mathcal{V}_1$ and $s_i = -1$ for $i \in \mathcal{V}_2$.

In order to analysis the BC control problem with coopetition networks, we define the local neighbour BC errors of each agent as follows

$$\varepsilon_i(k) = \sum_{j \in \mathcal{N}(i)} |a_{ij}|(x_i(k) - \text{sign}(a_{ij})x_j(k)) + b_i(x_i(k) - s_i x_0(k)). \tag{5}$$

Then, we define $\varepsilon(k) = (\varepsilon_1^\top(k), \ldots, \varepsilon_N^\top(k))^\top \in \mathcal{R}^{nN}$, $\eta_i(k) = x_i(k) - s_i x_0(k)$ as the BC error vector and the tracking error, respectively. Then, we can rewrite above local errors as following impact form

$$\varepsilon(k) = ((\mathcal{L} + \mathcal{B}) \otimes I_n)\eta(k), \tag{6}$$

where

$$\eta(k) = x - \bar{S}\bar{x}_0 \tag{7}$$

is the overall tracking error, $\bar{S} = S \otimes I_n$, $\bar{x}_0 = \mathbf{1} \otimes x_0$ and $\mathbf{1} = col(1, \ldots, 1) \in \mathcal{R}^N$ is the N-vector of ones, $x = (x_1^\top(k), x_2^\top(k), \ldots, x_N^\top(k))^\top \in \mathcal{R}^{nN}$.

Considering (6) and (7), from Lemma 1, the important relationship between the local neighbour BC errors $\varepsilon(k)$ and the tracking errors $\eta(k)$ can be obtained, namely, if $\lim_{k \to \infty} \|\varepsilon(k)\| = 0$, then $\lim_{k \to \infty} \|\eta(k)\| = 0$. So, once the local neighbour BC error decreases to 0, then the BC control problem can be achieve for all the agent according to Definition 1.

By combining (1), (2) and (5), for agent i, the dynamics of the local neighbour BC errors are given as follows

$$\varepsilon_i(k+1) = A\varepsilon_i(k) + (d_i + b_i)B_i u_i(k) - \sum_{j \in \mathcal{N}(i)} |a_{ij}|\text{sign}(a_{ij})B_j u_j(k). \tag{8}$$

In the field of RL [48], cost (reward) functions are usually employed to identify the performance of a series of control laws (actions). Inspired by it, for each agent, the discounted PIF which is defined as follows:

$$J_i(\varepsilon_i(k)) = \sum_{w=k}^{\infty} \alpha^{n-k} c_i(\varepsilon_i(w), u_i(w), u_{\mathcal{N}(i)}(w)), \tag{9}$$

where $c_i(\varepsilon_i(w), u_i(w), u_{\mathcal{N}(i)}(w))$ is the utility function, that is, $c_i(\varepsilon_i(w), u_i(w), u_{\mathcal{N}(i)}(w)) = \varepsilon_i^{\top}(w)Q_{ii}\varepsilon_i(w) + u_i^{\top}(w)R_{ii}u_i(w) + \sum_{j\in\mathcal{N}(i)} u_j^{\top}(w)S_{ij}u_j(w)$, $Q_{ii} > 0$, $R_{ii} > 0$, $S_{ij} > 0$ are symmetric positive matrices. $u_{\mathcal{N}(i)}$ denotes the sets of the control laws from the agent i's neighbors, and $\alpha \in (0, 1]$ is discount factor.

From (9), it is noted that the PIF for each agent i relies on the local neighbour BC error $\varepsilon_i(k)$, control input signal of itself $u_i(k)$ and its neighbors.

Definition 2 (OBCC problem)**.** The goal of the OBBC is to find control laws to guarantee BC control problem can be achieved in Definition 1 and minimize the PIF with respect to the local neighbour BC errors (5) at the same time.

Remark 2. It is noted that the general BC control problem for MASs has been investigated in [24,47]. Different from them, the goal of the paper is aim at designing the optimal control law $u_i(k)$ to ensure that not only the BC control can be achieved, but also the PIF (9) can be minimised, which increases the difficulty of the controller designs.

3. PI based OBCC design

This section employs the optimality principle and DT HJB equation to solve the OBCC problem while simultaneously minimize the PIF. We first derive the optimal control law according to the DT HJB equation and then obtain its approximate solution by using PIA.

3.1. DT HJB equation

In fact, for $\forall i$, the PIF (9) can be expressed as follows

$$J_i(\varepsilon_i(k), u_i(k), u_{\mathcal{N}(i)}(k)) = c_i(\varepsilon_i(k), u_i(k), u_{\mathcal{N}(i)}(k)) + \alpha J_i(\varepsilon_i(k+1), u_i(k+1), u_{\mathcal{N}(i)}(k+1)). \tag{10}$$

In the rest of the paper, for brevity, we let $J_i(\varepsilon_i(k), u_i(k), u_{\mathcal{N}(i)}(k)) = J_i(\varepsilon_i(k))$, $c_i(\varepsilon_i(k), u_i(k), u_{\mathcal{N}(i)}(k)) = c_i(\varepsilon_i(k), u_i(k))$, respectively.

In light of [38], and using Bellman's optimality principle, it can be obtained that the optimal PIF $J_i^*(\varepsilon_i(k))$ satisfies the coupled DT HJB, which is given by

$$J_i^*(\varepsilon_i(k)) = \min_{u_i(k)}\{c_i(\varepsilon_i(k), u_i(k)) + \alpha J_i^*(\varepsilon_i(k+1))\}. \tag{11}$$

The optimal control law satisfies $\partial J_i^*/\partial u_i = 0$, and therefore yields the optimal control law

$$u_i^*(k) = -\frac{\alpha}{2}(d_i + b_i)R_{ii}^{-1}B_i^{\top}\nabla J_i^*(\varepsilon_i(k+1)), \tag{12}$$

where $\nabla J_i^*(\varepsilon_i(k+1)) = \partial J_i^*(\varepsilon_i(k+1))/\partial\varepsilon_i(k+1)$.

Therefore, the OBCC can be solved based on the solution of the HJB equation. Unfortunately, the analytical solution of the HJB equation is generally impossible to be obtained. To overcome the above issue, an iterative method is employed to approximate the solution of the HJB Eq. (11) the next.

Remark 3. From (12), we can note that the above optimal controller design is a model-based pattern, which requires information of the explicit dynamics matrices B_i beforehand. However, this model-based method doesn't work for completely unknown systems in the practical scenarios. Therefore, to this end, a model-free-RL based method will be introduced to handle above difficulty in this paper.

3.2. PIA for the DT-HJB equation

In this section, a PIA is presented to estimate the approximate solution of the DT HJB equation. First, let $l \in [0, \infty)$ and $k \in [0, \infty)$ be the iteration index and time step, respectively. Then, let $u_i^l(k)$, $J_i^l(\varepsilon_i(k))$ be the iterative PIF and iterative control law, respectively.

PIA: Define the $u_i^0(k)$ as an initial admissible control law.

(1) (Policy evaluation): Update the performance index $J_i^l(\varepsilon_i(k))$ as follows:

$$J_i^l(\varepsilon_i(k)) = c_i(\varepsilon_i(k), u_i^l(k)) + \alpha J_i^l(\varepsilon_i(k+1)). \tag{13}$$

(2) (Policy improvement): The iterative control policy $u_i^l(k)$ for $\forall i$ is updated by:

$$u_i^{l+1}(k) = \arg\min_{u_i(k)}\{c_i(\varepsilon_i(k), u_i(k)) + \alpha J_i^l(\varepsilon_i(k+1))\}. \tag{14}$$

The iterations stop until $u_i^l(k)$ converges to $u_i^*(k)$.

It should be noted that the presented PIA can be mainly regarded as two parts: the policy evaluation part and the policy improvement parts. Repeat above process which stops when $u_i^l(k)$ is convergent. In the next section, the theoretical convergence analysis of the above PIA is provided.

4. Convergence analysis

In this section, the iterative $J_i^l(\varepsilon_i(k))$ and the iterative $u_i^l(k)$ will be proved that they can converge to the optimal value based on PIA, i.e. $J_i^*(\varepsilon_i(k))$ and $u_i^*(k)$, respectively, as $l\to\infty$. In the meantime, it is proved that the OBCC problem of the MASs can be guaranteed under the proposed optimal control laws.

Firstly, the following lemma will be used later for the proof of the convergence.

Lemma 2 [29]. *The iterative control laws $u_i^l(k)$ are also admissible laws for $l=0,1,2,\ldots$, if the initial control law $u_i^0(k)$ is the admissible control law.*

Theorem 1. *For $\forall i$, given the arbitrary initial admissible policies, compute the $J_i^l(\varepsilon_i(k))$ and the $u_i^l(k)$ by (13) and (14), respectively. Then, we have $J_i^l(\varepsilon_i(k))$ isn't monotonically increasing satisfying $J_i^{l+1}(\varepsilon_i(k))\le J_i^l(\varepsilon_i(k))$.*

Proof. This theorem will be proved by using mathematical induction. $\forall i$ and $\forall l$, a new iterative PIF $\Upsilon_i^{l+1}(\varepsilon_i(k))$ is defined as follows

$$\begin{aligned}\Upsilon_i^{l+1}(\varepsilon_i(k)) &\triangleq c_i(\varepsilon_i(k),u_i^{l+1}(k))+\alpha J_i^l(\varepsilon_i(k+1))\\ &= \min_{u_i(k)}\{c_i(\varepsilon_i(k),u_i(k))+\alpha J_i(\varepsilon_i(k+1))\}.\end{aligned}\tag{15}$$

Then, according to Eqs. (13)–(15), one has

$$\Upsilon_i^{l+1}(\varepsilon_i(k))\le J_i^l(\varepsilon_i(k)).\tag{16}$$

From Lemma 2, $u_i^{l+1}(k)$ is always admissible for arbitrary i and l. Therefore, it is noted that $\varepsilon_i(k)\to 0$ when $k\to\infty$. Let $k=q$ with $m\to\infty$, we can have

$$J_i^l(\varepsilon_i(q))=\Upsilon_i^{l+1}(\varepsilon_i(q))=J_i^{l+1}(\varepsilon_i(q)).\tag{17}$$

Let $k=q-1$, combining Eqs. (13), (14), and (17) yields

$$\begin{aligned}J_i^{l+1}(\varepsilon_i(q-1)) &= c_i(\varepsilon_i(q-1),u_i^{l+1}(q-1))+\alpha J_i^{l+1}(\varepsilon_i(q))\\ &= c_i(\varepsilon_i(q-1),u_i^{l+1}(q-1))+\alpha J_i^{l}(\varepsilon_i(q))\\ &= \min_{u_i(q-1)}\{c_i(\varepsilon_i(q-1),u_i(q-1))+\alpha Q_i^l(\varepsilon_i(q))\}\\ &\le c_i(\varepsilon_i(q-1),u_i^{l}(q-1))+\alpha J_i^l(\varepsilon_i(q))\\ &= J_i^l(\varepsilon_i(q-1)).\end{aligned}\tag{18}$$

It follows from (18) that the conclusion $J_i^{l+1}(\varepsilon_i(k))\le J_i^l(\varepsilon_i(k))$ holds when $k=q-1$. Next, we assume that it holds when $k=K+1$, $K=0,1,\ldots,$ that is,

$$J_i^{l+1}(\varepsilon_i(K+1))\le J_i^l(\varepsilon_i(K+1)).\tag{19}$$

Let $k=K$, we can obtain according to Eqs. (13), (14) and inequality (19) as follows

$$\begin{aligned}J_i^{l+1}(\varepsilon_i(K)) &= c_i(\varepsilon_i(K),u_i^{l}(K))+\alpha J_i^{l+1}(\varepsilon_i(K+1))\\ &\le c_i(\varepsilon_i(K),u_i^{l}(K))+\alpha J_i^{l}(\varepsilon_i(K+1))\\ &= \Upsilon_i^{l+1}(\varepsilon_i(K))\\ &\le J_i^l(\varepsilon_i(K)).\end{aligned}\tag{20}$$

Therefore, for arbitrary i and l, we conclude that $J_i^{l+1}(\varepsilon_i(k))\le J_i^l(\varepsilon_i(k))$ holds for arbitrary k according to inequality (20). Therefore, this completes the proof. □

Theorem 2. *For $\forall i$, given the arbitrary initial admissible control law and let $J_i^l(\varepsilon_i(k))$ and $u_i^l(k)$ be updated by (13) and (14). Then, we have that $J_i^l(\varepsilon_i(k))$ and $u_i^l(k)$ will converge to the optimum, when $l\to\infty$, that is,*

$$\lim_{l\to\infty}J_i^l(\varepsilon_i(k))=J_i^*(\varepsilon_i(k)),\ \lim_{l\to\infty}u_i^l(k)=u_i^*(k).$$

Proof. Firstly, the limit of the $J_i^l(\varepsilon_i(k))$ is denoted by $J_i^\infty(\varepsilon_i(k)) = \lim_{l\to\infty} J_i^l(\varepsilon_i(k))$. From Theorem 1, the $\{J_i^l(\varepsilon_i(k))\}$ isn't a monotonically increasing sequence, one has

$$\begin{aligned} J_i^\infty(\varepsilon_i(k)) \le\ & \Psi_i^{l+1}(\varepsilon_i(k)) \\ =\ & \min_{u_i(k)}\{c_i(\varepsilon_i(k), u_i(k)) + \alpha J_i^l(\varepsilon_i(k+1))\}. \end{aligned} \tag{21}$$

Then, let l be enough large, where $l\to\infty$, such that

$$J_i^\infty(\varepsilon_i(k)) \le \min_{u_i(k)}\{c_i(\varepsilon_i(k), u_i(k)) + \alpha J_i^\infty(\varepsilon_i(k+1))\}. \tag{22}$$

We chose χ as an arbitrary positive constant. Since $\{J_i^l(\varepsilon_i(k))\}$ isn't a monotonically increasing sequence, so it has a positive constant X yields

$$J_i^X(\varepsilon_i(k)) - \chi \le J_i^\infty(\varepsilon_i(k)) \le J_i^X(\varepsilon_i(k)). \tag{23}$$

Therefore, combine (23) and (10) yields

$$\begin{aligned} J_i^\infty(\varepsilon_i(k)) &\ge J_i^X(\varepsilon_i(k)) - \chi \\ &= c_i(\varepsilon_i(k), u_i^X(k)) + \alpha J_i^X(\varepsilon_i(k+1)) - \chi \\ &\ge c_i(\varepsilon_i(k), u_i^X(k)) + \alpha J_i^\infty(\varepsilon_i(k+1)) - \chi \\ &\ge \min_{u_i(k)}\{c_i(\varepsilon_i(k), u_i(k)) + \alpha J_i^\infty(\varepsilon_i(k+1))\} - \chi. \end{aligned} \tag{24}$$

Because χ is an arbitrary constant, one yields

$$J_i^\infty(\varepsilon_i(k)) \ge \min_{u_i(k)}\{c_i(\varepsilon_i(k), u_i(k)) + \alpha J_i^\infty(\varepsilon_i(k+1))\}. \tag{25}$$

From inequality (22) and (25), one has

$$J_i^\infty(\varepsilon_i(k)) = \min_{u_i(k)}\{c_i(\varepsilon_i(k), u_i(k)) + \alpha J_i^\infty(\varepsilon_i(k+1))\}. \tag{26}$$

Next, we define $v_i(k)$ as an arbitrary admissible policy, and $\Pi_i(\varepsilon_i(k))$ is defined as an another new PIF, which is given as follows

$$\Pi_i(\varepsilon_i(k)) = c_i(\varepsilon_i(k), v_i(k)) + \alpha\Pi_i(\varepsilon_i(k+1)). \tag{27}$$

Then, we set $k = s$, from Lemma 2, when $s\to\infty$, $\varepsilon_i(s)\to 0$, which gets $J_i^\infty(\varepsilon_i(s)) = \Pi_i(\varepsilon_i(s)) = 0$.
Let $k = s-1$, we can obtain

$$\begin{aligned} \Pi_i(\varepsilon_i(s-1)) =\ & c_i(\varepsilon_i(s-1), v_i(s-1)) + \alpha\Pi_i(\varepsilon_i(s)) \\ \ge\ & \min_{u_i(s-1)}\{c_i(\varepsilon_i(s-1), u_i(s-1)) + \alpha\Pi_i(\varepsilon_i(s))\} \\ =\ & \min_{u_i(s-1)}\{c_i(\varepsilon_i(s-1), u_i(s-1)) + \alpha J_i^\infty(\varepsilon_i(s))\} \\ =\ & J_i^\infty(\varepsilon_i(s-1)). \end{aligned} \tag{28}$$

Then, we assume the conclusion (28) holds for $k = p+1$ with $p\in\{0,1,2,\ldots\}$, that is $\Pi_i(\varepsilon_i(p+1)) \ge J_i^\infty(\varepsilon_i(p+1))$, thus when $k = p$, we have

$$\begin{aligned} \Pi_i(\varepsilon_i(p)) &= c_i(\varepsilon_i(p), v_i(p)) + \alpha\Pi_i(\varepsilon_i(p+1)) \\ &\ge c_i(\varepsilon_i(p), v_i(p)) + \alpha J_i^\infty(\varepsilon_i(p+1) \\ &= \min_{v_i(p)}\{c_i(\varepsilon_i(p), v_i(p)) + \alpha J_i^\infty(\varepsilon_i(p+1)) \\ &= J_i^\infty(\varepsilon_i(p)). \end{aligned} \tag{29}$$

Therefore, for $\forall k = 0,1,2,\ldots,$ it can be obtained that $\Pi_i(\varepsilon_i(k)) \ge J_i^\infty(\varepsilon_i(k))$. Let $v_i(k) = u_i^*(k)$ for $\forall i$, one has

$$J_i^\infty(\varepsilon_i(k) \le \Pi_i(\varepsilon_i(k)) = J_i^*(\varepsilon_i(k). \tag{30}$$

Since $J_i^*(\varepsilon_i(k)$ is the optimal value of the PIF, then we can get

$$J_i^\infty(\varepsilon_i(k)) \ge J_i^*(\varepsilon_i(k)). \tag{31}$$

Thus, combining inequality (30) and (31), it can be obtained that

$$\lim_{l\to\infty} J_i^l(\varepsilon_i(k)) = J_i^\infty(\varepsilon_i(k)) = J_i^*(\varepsilon_i(k)). \tag{32}$$

Therefore, the iterative PIF $J_i^l(\varepsilon_i(k))$ will converges to the optimal value $J_i^*(\varepsilon_i(k))$, as $l\to\infty$. Then, we can also obtain $\lim_{l\to\infty} u_i^l(k) = u_i^*(k)$ based on the optimal control law (12). Therefore, this completes the proof. □

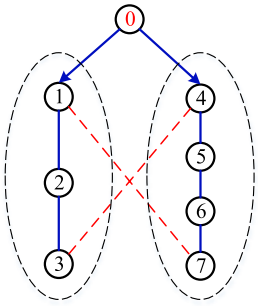


Fig. 1. Coopetition networks $\tilde{\mathcal{F}}$ with vertex set {0, 1, 2, 3, 4, 5, 6, 7}.

The next theorem gives the convergence proof that indicating that the OBCC problem can be achieved under coopetition network $\tilde{\mathcal{F}}$.

Theorem 3. *Assume that the Assumption 1 is held. For each agent i, if the optimal PIF $J_i^*(\varepsilon_i(k))$ satisfies the coupled DT HJB Eq. (11) and the optimal control law $u_i^*(k)$ is calculated according to (12). Thus, the local neighbour BC error $\varepsilon_i(k)$ is asymptotically stable, that is, $\lim_{k\to\infty}\varepsilon_i(k)=0$ and then OBCC problem can be achieved, that is, $\lim_{k\to\infty}\eta_i(k)=0$.*

Proof. Because the optimal PIF $J_i^*(\varepsilon_i(k))$ satisfies the DT HJB Eq. (11), we can obtain

$$c_i(\varepsilon_i(k),u_i^*(k))=J_i^*(\varepsilon_i(k))-\alpha J_i^*(\varepsilon_i(k+1)). \tag{33}$$

Then, we multiply both sides of (33) by α^k yields

$$\alpha^k c_i(\varepsilon_i(k),u_i^*(k))=\alpha^k J_i^*(\varepsilon_i(k))-\alpha^{k+1}J_i^*(\varepsilon_i(k+1)). \tag{34}$$

Next, We chose $\alpha^k J_i^*(\varepsilon_i(k))$ as Lyapunov function candidate, we can obtain

$$\triangle(\alpha^k J_i^*(\varepsilon_i(k)))=\alpha^{k+1}J_i^*(\varepsilon_i(k+1))-\alpha^k J_i^*(\varepsilon_i(k)). \tag{35}$$

Therefore, according to (34), the Eq. (35) can be rewritten as

$$\triangle(\alpha^k J^*(\varepsilon_i(k)))=-\alpha^k c_i(\varepsilon_i(k),u_i^*(k))<0.$$

From the above analysis, it is shown that the error system (8) is asymptotically stable, i.e., $\lim_{k\to\infty}\varepsilon_i(k)=0$. Then, according to Lemma 1 and the relationship between $\varepsilon_i(k)$ and $\eta_i(k)$, we can obtain that $\eta_i(k)\to 0$ as $k\to\infty$, thus, all the agent can achieve BC according to Definition 1. Further, the $J_i^*(\varepsilon_i(k))$ is the optimal vale of PIF, thus OBCC can be achieved by Definition 2 finally. This completes the proof. □

5. Model-free-RL based implementation for the OBCC

In this section, the NNs based structures, namely, actor-critic neural networks (AcNNs) [48], which are employed to implement the process of the presented OBCC method for each agent, that is, each agent has its own NNs architecture. The actor NNs is established to estimate the control laws, the critic NNs is regarded as the approximator of the PIF, respectively. In this paper, 3-layer NNs are selected for AcNNs. The detailed establishment process of the AcNNs are described as follows.

5.1. The actor-critic NNs

First, the critic NN is used to approximate the PIF of each agent. Thus, the approximate value of the PIF for each agent i is represented as

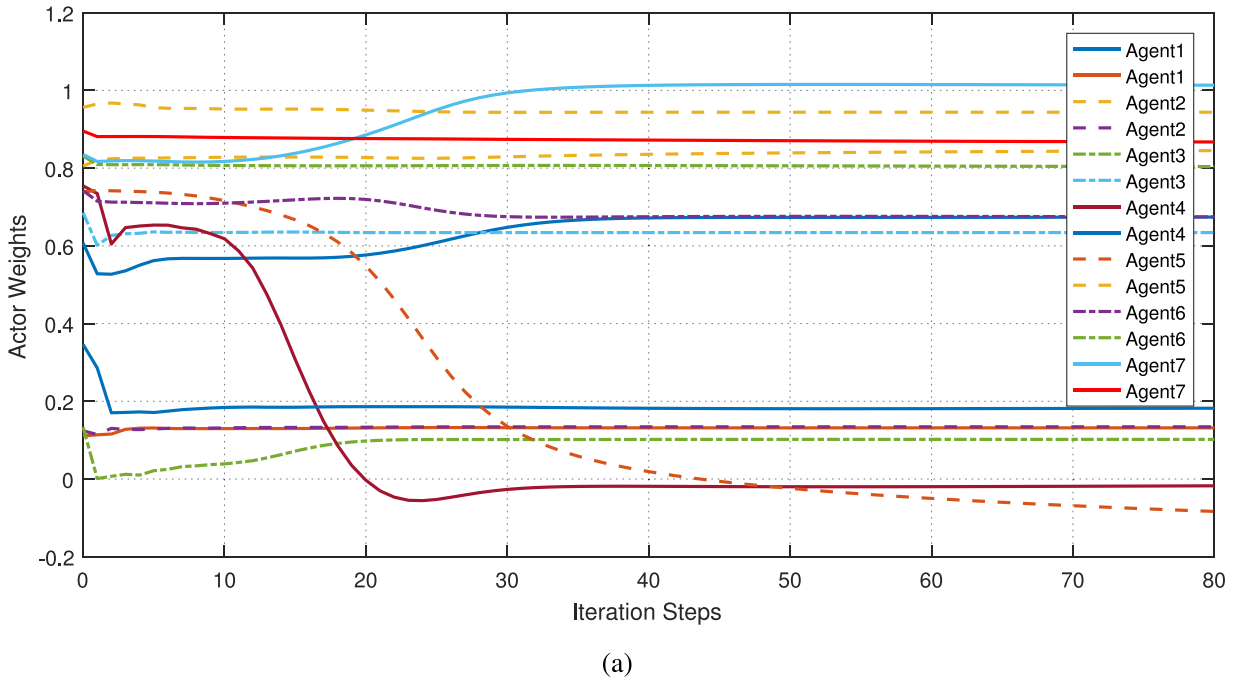
$$\hat{J}_i(k)=w_{c2i}^{\top}f(w_{c1i}^{\top}z_{ci}(k)), \tag{36}$$

where $z_{ci}(k)$ denotes the input information containing $\varepsilon_i(k)$, $u_i(k)$ and $u_{\mathcal{N}(i)}(k)$; And w_{c1i} represents input-to-hidden (ITH) layer weights vector, w_{c2i} represents the hidden-to-output (HTO) layer weight vector, and $f(\cdot)$ denotes the activation function for the critic networks. The activation function is expressed by the sigmoid function, i.e.,

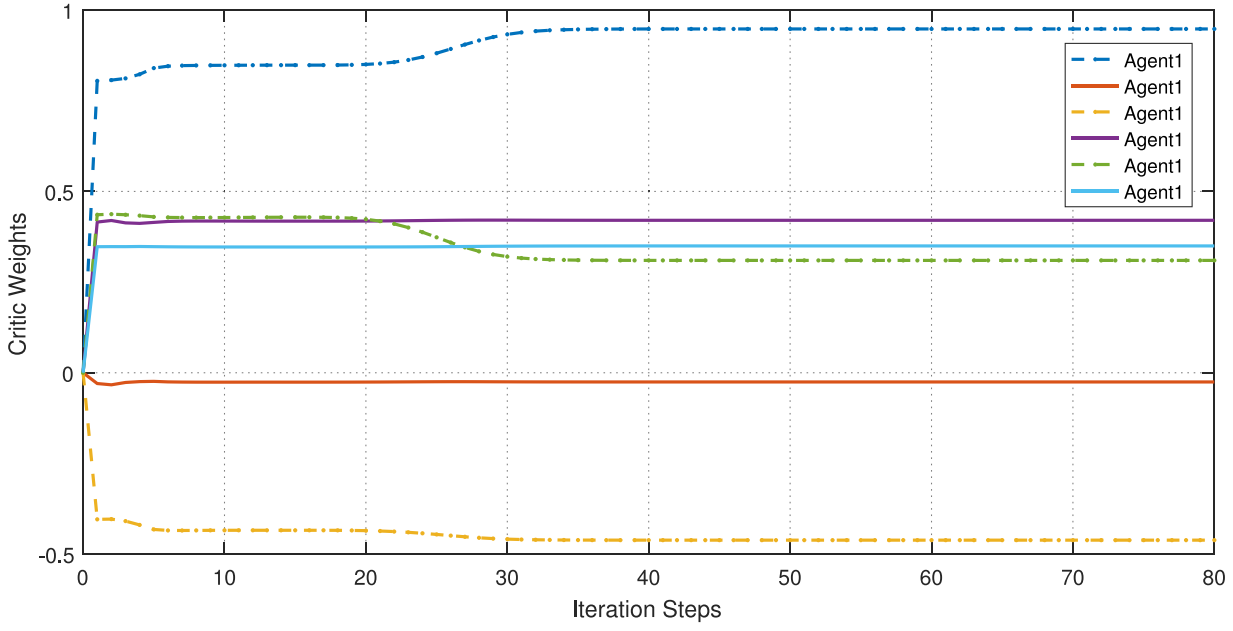
$$f(x)=\frac{1-e^{-x}}{1+e^{-x}}. \tag{37}$$

Then, the approximate error function for critic NN is given by

$$e_{ci}(k)=c_i(k)+\alpha\hat{J}_i(k+1)-\hat{J}_i(k). \tag{38}$$



(a)



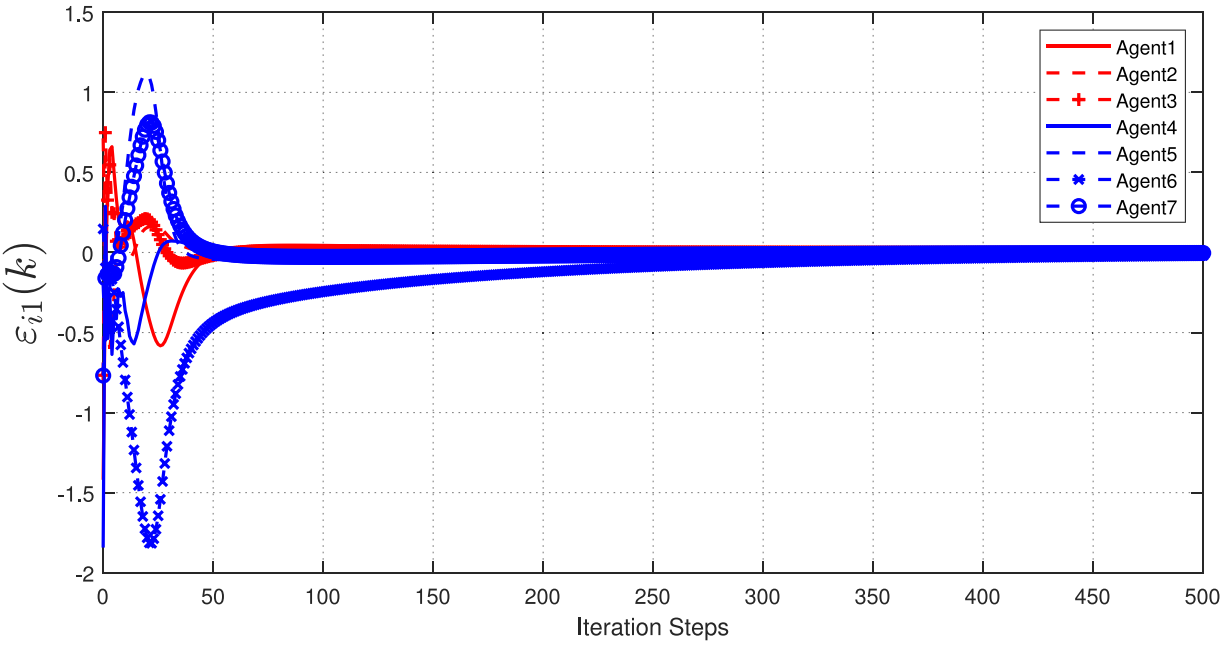
(b)

Fig. 2. (a) The actor NNs weight elements of all agents ($w_{a2i}^{(l)}, i = 1, 2, 3, 4, 5, 6, 7$); (b) The critic NNs weight elements of agent 1 ($w_{c21}^{(l)}$).

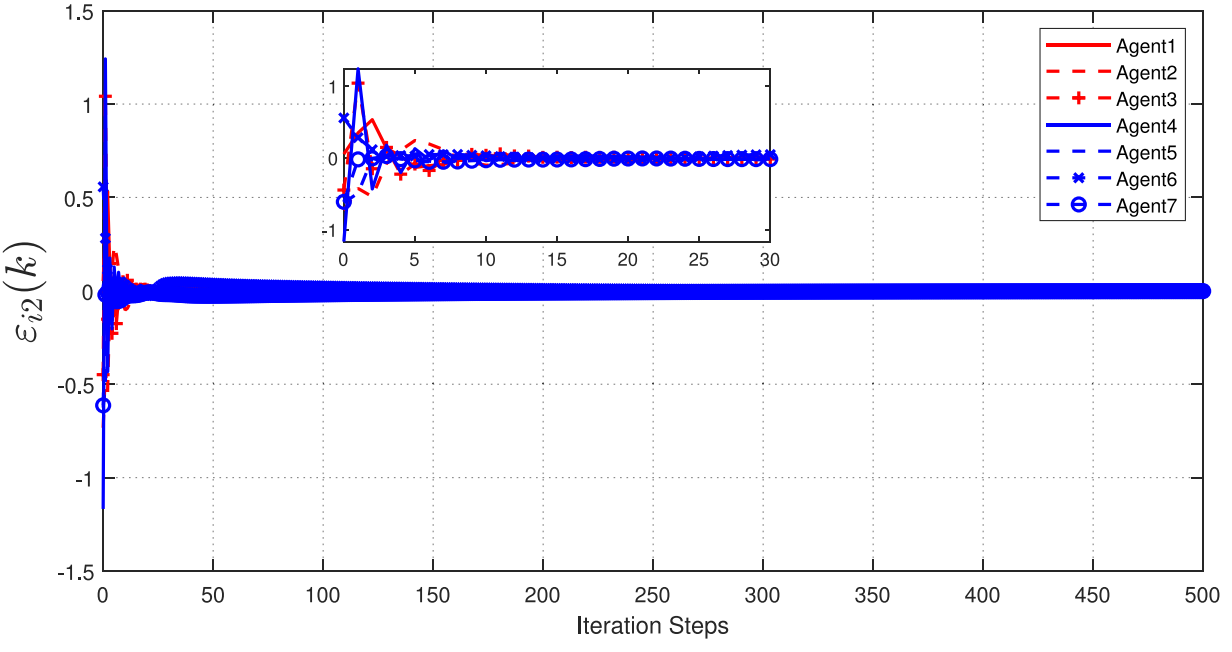
In the critic NN, we define an objective loss function which will be minimized is

$$l_{ci}(k) = \frac{1}{2} e_{ci}(k)^2.$$

Herein, for the sake of convenience, the HTO weight is chosen as an unit matrix, that is, $w_{c1i} = I$. Then, the gradient-descent-rule [49] (GDR) is utilized to tuning the HTO weight parameters, which is given as follows



(a)



(b)

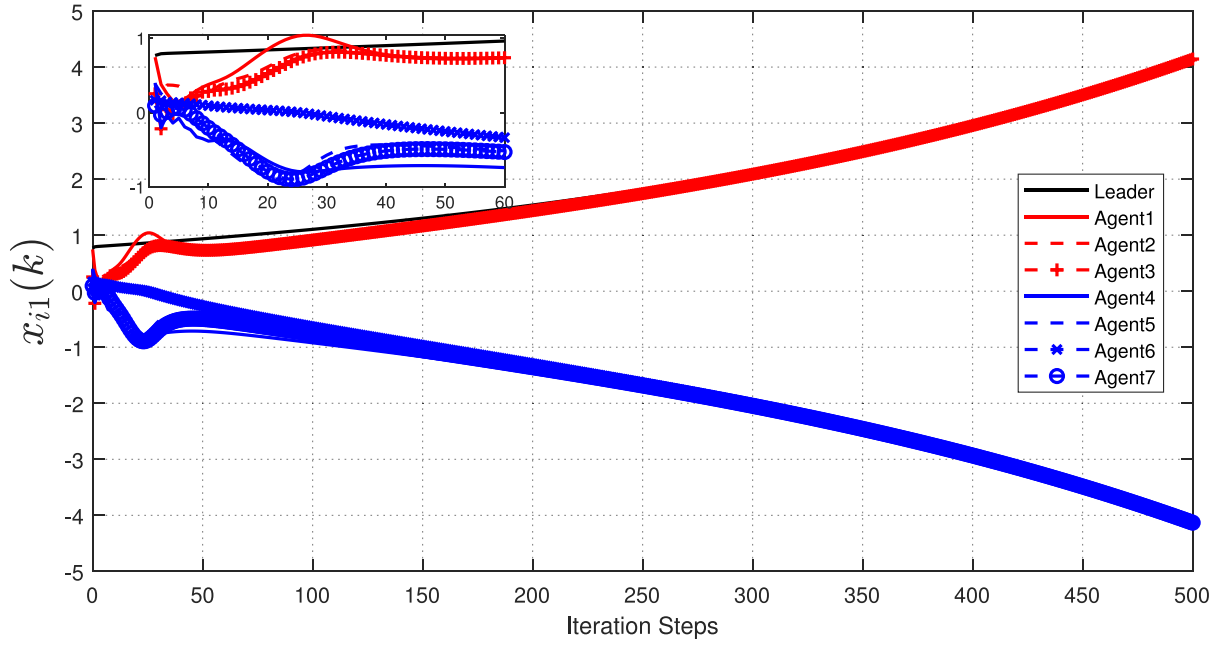
Fig. 3. (a) Evolution of the local neighbour BC errors $\varepsilon_{i1}(k)$ $(i = 1, 2, 3, 4, 5, 6, 7)$; (b) Evolution of the local neighbour BC errors $\varepsilon_{i2}(k)$ $(i = 1, 2, 3, 4, 5, 6, 7)$.

$$w_{c2i}^{(l+1)} = w_{c2i}^{(l)} - \kappa_{ci} \frac{\partial l_{ci}(k)}{\partial e_{ci}(k)} \frac{\partial e_{ci}(k)}{\partial w_{c2i}(k)}, \tag{39}$$

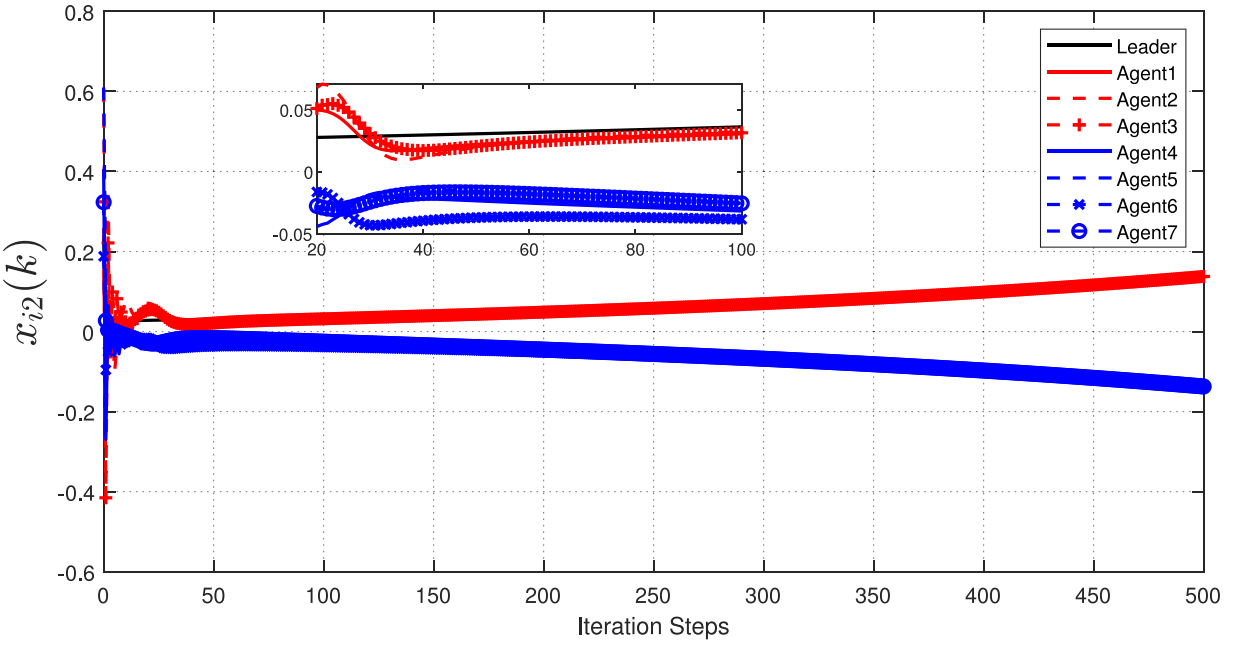
where κ_{ci} is regarded as the learning rate for the critic NNs.

Next, in order to approximate the optimal control law, the actor NNs is presented, where the output of the actor NNs is represented as

$$\hat{u}_i(k) = w_{a2i}^{\top} f(w_{a1i}^{\top} z_{ai}(k)), \tag{40}$$



(a)



(b)

Fig. 4. (a) Evolution of state of each agent's first coordinate $x_{i1}(k)$ $(i = 0, 1, 2, 3, 4, 5, 6, 7)$; (b) Evolution of state of each agent's second coordinate $x_{i2}(k)$ $(i = 0, 1, 2, 3, 4, 5, 6, 7)$.

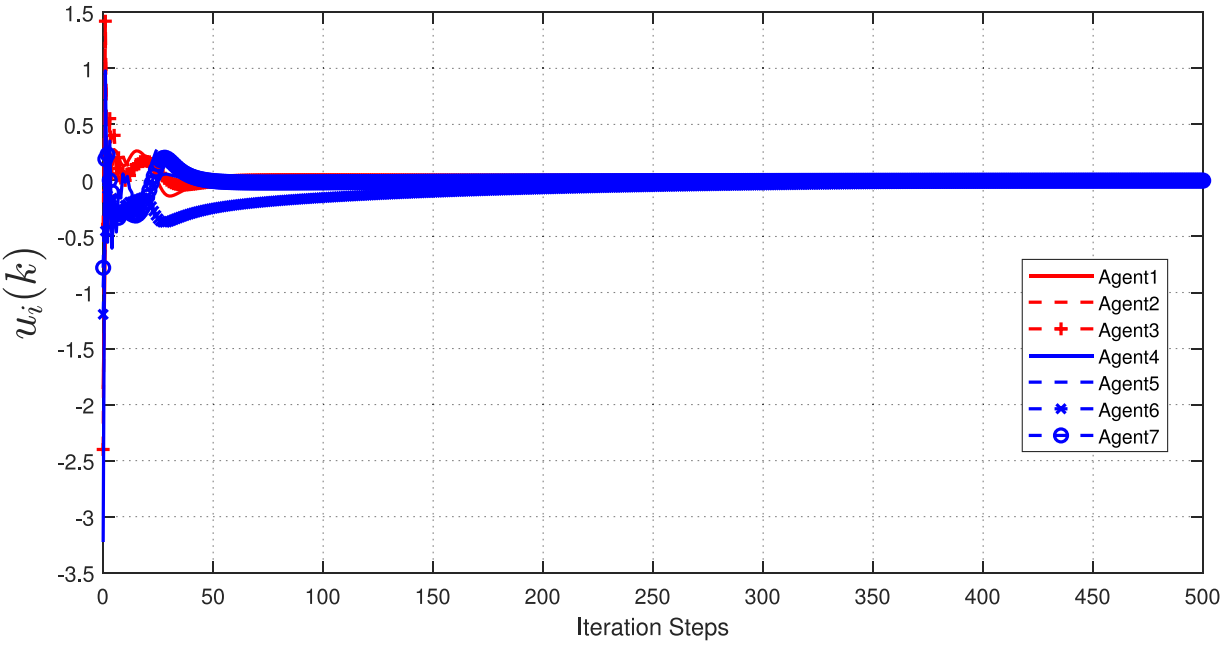


Fig. 5. The trajectories of the control laws $u_i(k)$ $(i = 1, 2, 3, 4, 5, 6, 7)$.

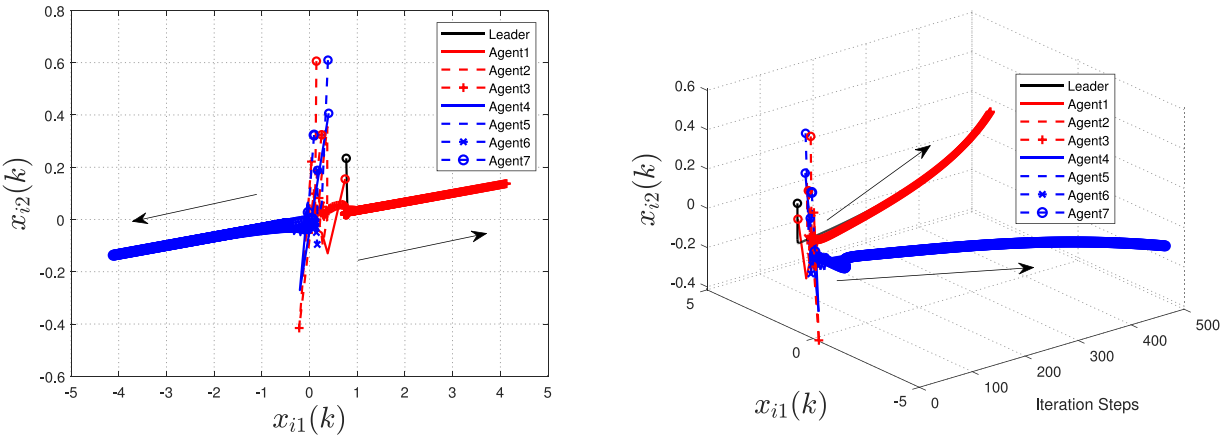


Fig. 6. 3-D and 2-D phase plane plots of the states for seven agents and one leader.

where $f(\cdot)$ is activation function for the actor NNs. The activation function is the same definition as above. Let $z_{ai}(k)$ be the input information for actor NNs consisting $\varepsilon_i(k)$. w_{a1i} is the weights of ITH and w_{a2i} represents the HTO layer.

Next the actor's network error function is defined as follows

$$e_{ai}(k) = \hat{J}_i(k) - U_i, \tag{41}$$

where U_i is the cost-to-go objective which is designed to be zero. Then, in order to updating the actor network, the objective function is defined as

$$l_{ai}(k) = \frac{1}{2} e_{ai}^2(k). \tag{42}$$

As above, we keep the ITH weight of actor network as constant, that is, $W_{a1i} = I$. Then, according to GDR, the weight of HTO is updated as follows

$$w_{a2i}^{(l+1)} = w_{a2i}^{(l)} - \kappa_{ai} \frac{\partial l_{ai}(k)}{\partial e_{ai}(k)} \frac{\partial e_{ai}(k)}{\partial \hat{J}_i(k)} \frac{\partial \hat{J}_i(k)}{\partial \hat{u}_i(k)} \frac{\partial \hat{u}_i(k)}{\partial w_{a2i}(k)}, \tag{43}$$

where κ_{ai} is the learning rate for the actor network.

5.2. Model-free-RL online learning scheme

Notice that on the basis of the AcNNs structure and the GDR of weight updating, the whole algorithmic framework of the model-free-RL based OBCC method is presented in Algorithm 1. It should be emphasized out that only system data (x_i, x_0, u_i) are involved in the process of the AcNNs framework to obtain the optimal control laws rather than the explicit SDs. That is the system matrices A and B_i are not appeared in the whole learning process of the dual-networks. Therefore, it is said that the proposed method is a model-free controller designs.

Algorithm 1 Model-free-RL based OBCC algorithm.

Initialization: (For each agent i)

Initialize followers $x_i(0)$ and leader $x_0(0)$, respectively;
Computation precision ϵ;
Calculate the initial local neighbour BC error $\varepsilon_i(0) \leftarrow$ (5);
Initialize the critic weights $w_{c2i}^{(0)}$ with zero values and initialize the actor weights $w_{a2i}^{(0)}$ in [0,1] randomly;
κ_{ai} and κ_{ci} are learning rates;
Set Q_i, R_i and S_{ij} as positive definite matrices;

Iteration:

Let the iteration and step index $k = 0$, $l = 0$;
repeat
  The control law $\hat{u}_i(k) \leftarrow$ (42);
  The local performance indices $\hat{J}_i(k) \leftarrow$ (38);
  The bipartite error $\varepsilon_i(k+1) \leftarrow$ (5) according to available system data $x_i(k+1)$ and $x_0(k+1)$;
  The control law $\hat{u}_i(k+1) \leftarrow$ (42);
  The performance index functions $\hat{J}_i(k+1) \leftarrow$ (38);
  Update the critic network weight $w_{c2i}^{(l+1)}$ by

$$w_{c2i}^{(l+1)} \leftarrow w_{c2i}^{(l)} - \kappa_{ci}[c_i(k) + \alpha\hat{J}_i(k+1) - \hat{J}_i(k)][\alpha f(\hat{z}_{ci}(k+1)) - f(\hat{z}_{ci}(k))] \tag{44}$$

  where $\hat{z}_{ci}(k) = w_{c1i}^{\top} z_{ci}(k)$ and κ_{ci} is the learning rate.
  Update the actor network weight $w_{a2i}^{(l+1)}$ by

$$w_{a2i}^{(l+1)} \leftarrow w_{a2i}^{(l)} - \kappa_{ai} w_{c2i}^{\top}(k) f(\hat{z}_{ci}(k)) w_{c2i} \psi_{ci}^{'} C_i f(\hat{z}_{ai}(k)) \tag{45}$$

  where $\hat{z}_{ai}(k) = w_{a1i}^{\top} z_{ai}(k)$, $\psi_{ci}^{'} = \partial f(\hat{z}_{ci}(k))/\partial \hat{z}_{ci}(k)$ and $C_i = \partial \hat{z}_{ci}(k)/\partial \hat{u}_i(k)$, and κ_{ai} is the learning rate.
  Let $l = l + 1$, $k = k + 1$;
until $\sum_{i=1}^{N} \|w_{c2i}^{l+1} - w_{c2i}^{l+1}\| \le \epsilon$;
return w_{a2i}, w_{c2i}, End.

6. Simulation results

In this section, we conduct experiment on numerical simulation environment to evaluate the effectiveness of the proposed approach.

We consider a LF MASs with 1 leader and 7 followers. The interaction network $\tilde{\mathcal{F}}$ among agents is shown in Fig. 1, where nodes 1,2,3,4,5,6,7 denote the follower agents, and the node 0 denotes the leader. From Fig. 1, the agents can be divided into two competitive subgroups (agents $1-3$ belong to the subgroup $\mathcal{V}_1$, agents $4-7$ belong to the subgroup $\mathcal{V}_2$). The *cooperative* relationships among agents are expressed by the blue solid lines, and the *competitive* relationships are denoted as red dashed lines. Notice that agent 1 and agent 4 can receive the information from the leader 0. The leader can be treated as a coordinator for the two subgroups. From the coopetition network $\mathcal{F}$, we can obtain the adjacency matrix $\boldsymbol{A}_{\mathcal{F}}$ with non-zero elements $a_{12} = a_{21} = a_{23} = a_{32} = a_{45} = a_{54} = a_{56} = a_{65} = a_{67} = a_{54} = 1$, $a_{17} = a_{71} = a_{34} = a_{43} = -1$, leader adjacency matrice is $\mathcal{B} = diag\{1, 0, 0, 1, 0, 0, 0\}$.

The SDs of the followers and the leader are the same as (1) and (2), respectively. The system matrices are given by $A = \begin{bmatrix} 1 & 0.1 \\ 0.03 & 0.1 \end{bmatrix}$, $B_1 = \begin{bmatrix} 0.2047 \\ 0.0898 \end{bmatrix}$, $B_2 = \begin{bmatrix} 0.2147 \\ 0.2895 \end{bmatrix}$, $B_3 = \begin{bmatrix} 0.2097 \\ 0.1897 \end{bmatrix}$. $B_4 = \begin{bmatrix} 0.2 \\ 0.1 \end{bmatrix}$. $B_5 = \begin{bmatrix} 0.2 \\ 0.01 \end{bmatrix}$. $B_6 = \begin{bmatrix} 0.02 \\ 0.1 \end{bmatrix}$. $B_7 = \begin{bmatrix} 0.2 \\ 0.01 \end{bmatrix}$.

The weighting matrices are selected as $Q_{11} = Q_{22} = Q_{33} = Q_{44} = Q_{55} = Q_{66} = Q_{77} = I_{2\times 2}$, $R_{11} = R_{22} = R_{33} = R_{44} = R_{55} = R_{66} = R_{77} = 1$, $S_{21} = S_{31} = S_{42} = S_{12} = S_{13} = S_{14} = S_{23} = S_{24} = S_{32} = S_{34} = S_{41} = S_{43} = 1$. For the implement of the proposed method, each agent has its own AcNNs in the learning process. And they only depend on local information when network learning. Let the discount factor $\alpha = 0.95$, and the learning rates are selected as $\kappa_{ci} = \kappa_{ai} = 0.05$ for $\forall i = 1, 2, 3, 4, 5, 6, 7$. The computation precision is $\epsilon = 10^{-6}$. In addition, we set initial critic weights $w_{c2i}^{(0)}$ $(i = 1, 2, 3, 4, 5, 6, 7)$ as zero and the initial actor weights $w_{a2i}^{(0)}$ $(i = 1, 2, 3, 4, 5, 6, 7)$ are randomly initialized in [0,1]. The initial state of the leader and the follow-

ers are chosen as $x_0(0) = [0.7696, 0.2341]^\top$, $x_1(0) = [0.7461, 0.1548]^\top$, $x_2(0) = [0.1439, 0.6060]^\top$, $x_3(0) = [0.2545, 0.3242]^\top$, $x_4(0) = [0.4018, 0.4064]^\top$, $x_5(0) = [0.3862, 0.6098]^\top$, $x_6(0) = [0.1669, 0.1881]^\top$, and $x_7(0) = [0.0946, 0.3232]^\top$.

Fig. 2 (a) shows the weight parameters of the actor NNs for all agents are finally convergent. The update process of the weight parameters of the critic NNs for agent 1 is depicted in Fig. 2 (b). The trajectories of the neighborhood BC errors ε_{i1} and ε_{i2} are shown in Fig. 3, respectively. The state trajectories of all seven agents and the leader are shown in Fig. 4 (a) and (b), which is shown that the leader can intervene in the two competitive subgroups effectively and thus guarantee the two subgroups to reach BC with respect to the state of the leader finally. The trajectories evolution process of control laws for all agents are given in Fig. 5. The 2-D and 3-D phase plane plots of the trajectory of the states x_i for all agents are shown in Fig. 6 (a) and Fig. 6 (b), respectively. It is shown that the seven agents reach BC which confirms the performance of the presented approaches.

7. Conclusions

In this paper, the OBCC problem of MASs with unknown dynamics has been investigated using model-free-RL method. In contrast to the traditional BC control methods with requiring the completely SDs, our proposed approach not only makes BC control achieved without needing the explicit system models, but also the PIFs can be minimized. The coopetition networks have been applied to model the cooperative-competitive interactions among agents. The local neighbour BC errors and PIFs have been defined for each agents to derive the DT HJB equations and the optimal control laws. Then, the approximate optimal control solutions have been obtained for each agent by utilizing the PIA. Further, the AcNNs mechanism and the gradient descent rule have been used to implement the proposed controller designs in an online learning manner, where the system models are no longer needed. Finally, several numerical simulation results have justified the effectiveness of the presented approaches. In the future work, the designed OBCC scheme will be extended to TS Fuzzy Systems and discrete-time systems [50,51].

Acknowledgments

This work is partially supported by National Science Foundation of China under Grants Nos. 61703060,61473061, 61104104, the Opening Fund of Geomathematics Key Laboratory of Sichuan Province (scsxdz2018zd02 and scsxdz2018zd04), the Fundamental Research Funds for the Central Universities, Southwest Minzu University (2019NQN07) and the Program for New Century Excellent Talents in University under Grant No. NCET-13-0091.

References

[1] B.D.O. Anderson, B. Fidan, C. Yu, D. Walle, UAV Formation control: Theory and application, Springer, London, 2008.
[2] G. Wen, X. Yu, Z. Liu, W. Yu, Adaptive consensus-based robust strategy for economic dispatch of smart grids subject to communication uncertainties, IEEE Trans. Ind. Informat. 14 (6) (2018) 2484–2496.
[3] X. Dong, B. Yu, Z. Shi, Y. Zhong, Time-varying formation control for unmanned aerial vehicles: theories and applications, IEEE Trans. Control Syst. Technol. 23 (1) (2015) 340–348.
[4] P.K.C. Wang, Navigation strategies for multiple autonomous mobile robots moving in formation, J. Robotic Syst. 8 (2) (2010) 177–195.
[5] X. Liu, K. Zhang, W. Xie, Pinning impulsive synchronization of reaction-diffusion neural networks with time-varying delays, IEEE Trans. Neural Netw. Learn. Syst. 28 (5) (2017) 1055–1067.
[6] H. Shen, Y. Zhu, L. Zhang, J.H. Park, Extended dissipative state estimation for Markov jump neural networks with unreliable links, IEEE Trans. Neural Netw. Learn. Syst. 28 (2017) 346–358.
[7] Z. Yu, H. Jiang, X. Mei, C. Hu, Guaranteed cost consensus for second-order multi-agent systems with heterogeneous inertias, Appl. Math. Comput. 338 (2018) 739–757.
[8] J. Tao, Z.G. Wu, H.Y. Su, Y.Q. Wu, D. Zhang, Asynchronous and resilient filtering for Markovian jump neural networks subject to extended dissipativity, IEEE Trans. Cyber. 49 (7) (2019) 2504–2513.
[9] K. Shi, J. Wang, S. Zhong, X. Zhang, Y. Liu, J. Cheng, New reliable nonuniform sampling control for uncertain chaotic neural networks under Markov switching topologies, Appl. Math. Comput. 347 (2019) 169–193.
[10] R. Olfatisaber, R.M. Murray, Consensus problems in networks of agents with switching topology and time-delays, IEEE Trans. Autom. Control 49 (9) (2004) 1520–1533.
[11] W. Ren, R.W. Beard, Consensus seeking in multiagent systems under dynamically changing interaction topologies, IEEE Trans. Autom. Control 50 (5) (2005) 655–661.
[12] Y. Hong, J. Hu, L. Gao, Tracking control for multi-agent consensus with an active leader and variable topology, Automatica 42 (7) (2006) 1177–1182.
[13] G. Shi, Y. Hong, Global target aggregation and state agreement of nonlinear multi-agent systems with switching topologies, Automatica 45 (5) (2009) 1165–1175.
[14] J. Hu, Y. Hong, Leader-following coordination of multi-agent systems with coupling time delays, Phys. A Stat. Mech. Appl. 374 (2) (2007) 853–863.
[15] D. Ye, M. Chen, H. Yang, Distributed adaptive event-triggered fault-tolerant consensus of multiagent systems with general linear dynamics, IEEE Trans. Cybern. 49 (3) (2019) 757–767.
[16] D. Ye, X. Yang, L. Su, Fault-tolerant synchronization control for complex dynamical networks with semi-Markov jump topology, Appl. Math. Comput. 312 (2017) 36–48.
[17] W.H. Riker, The Theory of Political Coalitions, Yale University Press, New Haven, Conn, USA, 1962.
[18] A. Ware, The Dynamics of Two-Party Politics: Party Structures and the Management of Competition, Comparative Politics, Oxford University Press, Oxford, UK, 2009.
[19] C. Altafini, Consensus problems on networks with antagonistic interactions, IEEE Trans. Autom. Control 58 (2013) 935–946.
[20] J. Hu, H. Zhu, Adaptive bipartite consensus on coopetition networks, Phys. D 307 (2015) 14–21.
[21] D. Meng, M. Du, Y. Jia, Interval bipartite consensus of networked agents associated with signed digraphs, IEEE Trans. Autom. Control 61 (12) (2016) 3755–3770.
[22] H. Ma, D. Liu, D. Wang, B. Lou, Bipartite output consensus in networked multi-agent systems of high-order power integrators with signed digraph and input noises, Int. J. Syst. Science 47 (13) (2016) 3116–3131.

[23] Y. Wu, L. Liu, J. Hu, G. Feng, Adaptive antisynchronization of multi-layer reactioncdiffusion neural networks, IEEE Trans. Neural Netw. and Learn. Syst. 29 (4) (2018) 807–818.
[24] J. Hu, Y. Wu, L. Liu, G. Feng, Adaptive bipartite consensus control of high-order multiagent systems on coopetition networks, Int. J. Robust Nonlin. Control 28 (7) (2018) 2868–2886.
[25] J. Hu, W. Zheng, Emergent collective behaviors on coopetition networks, Phys. Lett. A 378 (26–27) (2014) 1787–1796.
[26] S. Shamshirband, A. Patel, N.B. Anuar, M.L.M. Kiah, A. Abraham, Cooperative game theoretic approach using fuzzy q-learning for detecting and preventing intrusions in wireless sensor networks, Eng. Appl. Artif. Intell. 32 (2014) 228–241.
[27] A. Kalantari, A. Kamsin, S. Shamshirband, A. Gani, H. Alinejad-Rokny, A.T. Chronopoulos, Computational intelligence approaches for classification of medical data: state-of-the-art, future challenges and research directions, Neurocomputing 276 (2018) 2–22.
[28] F. Fotovatikhah, M. Herrera, S. Shamshirband, K.W. Chau, S.F. Ardabili, M.J. Piran, Survey of computational intelligence as basis to big flood management: challenges, research directions and future work, Eng. Appl. Comp. Fluid Mech. 12 (1) (2018) 411–437.
[29] M. Abu-Khalaf, F.L. Lewis, Nearly optimal control laws for nonlinear systems with saturating actuators using a neural network hjb approach, Automatica 41 (5) (2005) 779–791.
[30] S. Bhasin, R. Kamalapurkar, M. Johnson, K.G. Vamvoudakis, F.L. Lewis, W.E. Dixon, A novel actor-critic-identifier architecture for approximate optimal control of uncertain nonlinear systems, Automatica 49 (1) (2013) 82–92.
[31] Z. Jiang, Y. Jiang, Robust adaptive dynamic programming for linear and nonlinear systems: an overview, Eur. J. Control 19 (5) (2013) 417–425.
[32] K.G. Vamvoudakis, H. Ferraz, Model-free event-triggered control algorithm for continuous-time linear systems with optimal performance, Automatica 87 (2018) 412–420.
[33] D. Liu, Q. Wei, Policy iteration adaptive dynamic programming algorithm for discrete-time nonlinear systems, IEEE Trans. Neural Netw. Learn. Syst. 25 (3) (2014) 621–634.
[34] A. Al-Tamimi, F.L. Lewis, M. Abu-Khalaf, Discrete-time nonlinear hjb solution using approximate dynamic programming: convergence proof, IEEE Trans. Syst. Man Cyber. Part B Cyber. 38 (4) (2008) 943–949.
[35] H. Zhang, Q. Wei, Y. Luo, A novel infinite-time optimal tracking control scheme for a class of discrete-time nonlinear systems via the greedy hdp iteration algorithm, IEEE Trans. Syst. Man Cyber. Part B Cyber. 38 (4) (2008) 937–942.
[36] D. Wang, D. Liu, Q. Wei, D. Zhao, N. Jin, Optimal control of unknown nonaffine nonlinear discrete-time systems based on adaptive dynamic programming, Automatica 48 (8) (2012) 1825–1832.
[37] J.J. Murray, C.J. Cox, G.G. Lendaris, R. Saeks, Adaptive dynamic programming, IEEE Trans. Syst. Man Cyber. Part C: Appl. Rev. 32 (2) (2002) 140–153.
[38] M.I. Abouheaf, F.L. Lewis, K.G. Vamvoudakis, S. Haesaert, R. Babuska, Multi-agent discrete-time graphical games and reinforcement learning solutions, Automatica 50 (12) (2014) 3038–3053.
[39] 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 (5) (2017) 4091–4100.
[40] Z. Peng, Y. Zhao, J. Hu, B.K. Ghosh, Data-driven optimal tracking control of discrete-time multi-agent systems with two-stage policy iteration algorithm, Inf. Sci. 481 (2019) 189–202.
[41] X. Zhong, H. He, Grhdp solution for optimal consensus control of multiagent discrete-time systems, IEEE Trans. Syst. Man Cyber. Syst. (2018), doi:10.1109/TSMC.2018.2814018.
[42] X. Bu, Z. Hou, H. Zhang, Data-driven multiagent systems consensus tracking using model free adaptive control, IEEE Trans. Neural Netw. Learn. Syst. 29 (5) (2018) 1514–1524.
[43] H. Zhang, D. Yue, C. Dou, W. Zhao, X. Xie, Data-driven distributed optimal consensus control for unknown multiagent systems with input-delay, IEEE Trans. Cyber. 49 (6) (2019) 2095–2105.
[44] J. Li, H. Modares, T. Chai, F.L. Lewis, L. Xie, Off-policy reinforcement learning for synchronization in multiagent graphical games, IEEE Trans. Neural Netw. Learn. Syst. 28 (10) (2017) 2434–2445.
[45] Z. Peng, J. Hu, B.K. Ghosh, Data-driven containment control of discrete-time multi-agent systems via value iteration, Sci. China Inf. Sci. (2018), doi:10.1007/s11432-018-9671-2.
[46] Z. Peng, J. Zhang, J. Hu, R. Huang, B.K. Ghosh, Optimal containment control of continuous-time multi-agent systems with unknown disturbances using data-driven approach, Sci. China Inf. Sci. (2019), doi:10.1007/s11432-019-9868-2.
[47] Y. Wu, J. Hu, Y. Zhang, Y. Zeng, Interventional consensus for high-order multi-agent systems with unknown disturbances on coopetition networks, Neurocomputing 194 (2016) 126–134.
[48] F.L. Lewis, D. Liu, Reinforcement Learning and Approximate Dynamic Programming for feedback Control, Wiley, New York, NY, USA, 2013.
[49] J. Si, Y.T. Wang, Online learning control by association and reinforcement, IEEE Trans. Neural Netw. 12 (2) (2001) 264–276.
[50] X.H. Chang, G.H. Yang, Nonfragile h_∞ filter design for TS fuzzy systems in standard form, IEEE Trans. Ind. Electron. 7 (61) (2014) 3448–3458.
[51] X.H. Chang, R. Huang, J.H. Park, Obust guaranteed cost control under uigital communication channels, IEEE Trans. Ind. Inf. (2019), doi:10.1109/TII.2019.2916146.