Neurocomputing 413 (2020) 85–95

Contents lists available at ScienceDirect

Neurocomputing

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

Internal reinforcement adaptive dynamic programming for optimal containment control of unknown continuous-time multi-agent systems

Jiefu Zhang [a], Zhinan Peng [a], Jiangping Hu [a,*], Yiyi Zhao [b], Rui Luo [a], Bijoy Kumar Ghosh [a,c]

[a] School of Automation Engineering, University of Electronic Science and Technology of China, Chengdu 611731, PR China
[b] School of Business Administration, Southwestern University of Finance and Economics, Chengdu 611130, PR China
[c] Department of Mathematics and Statistics, Texas Tech University, Lubbock, TX 79409-1042, USA

ARTICLE INFO

Article history:
Received 9 September 2019
Revised 3 May 2020
Accepted 26 June 2020
Available online 6 July 2020
Communicated by Huaguang Zhang

Keywords:
Optimal containment control
Multi-agent system
Internal reinforcement learning
Adaptive dynamic programming
Neural network

ABSTRACT

In this paper, a novel control scheme is developed to solve an optimal containment control problem of unknown continuous-time multi-agent systems. Different from traditional adaptive dynamic programming (ADP) algorithms, this paper proposes an internal reinforcement ADP algorithm (IR-ADP), in which the internal reinforcement signals are added in order to facilitate the learning process. Then a distributed containment control law is designed for each agent with the internal reinforcement signal. The convergence of this IR-ADP algorithm and the stability of the closed-loop multi-agent system are analyzed theoretically. For the implementation of the optimal controllers, three neural networks (NNs), namely internal reinforcement NNs, critic NNs and actor NNs, are utilized to approximate the internal reinforcement signals, the performance indices and optimal control laws, respectively. Finally, some simulation results are provided to demonstrate the effectiveness of the proposed algorithm.

1. Introduction

In the past decades, distributed control of multi-agent system (MAS) has attracted much attention by the communities of computer science, control theory and energy engineering. A MAS consists of a group of autonomous agents which interact with each other under communication topologies [1] can be used to describe a wide variety of complex systems such as power systems [2,3], sensor networks [4], spacecraft systems [5,6] and robotic systems [7,8]. Various interesting problems, including leader–follower tracking problems [9,10] and consensus problems [11–14], have been investigated extensively due to their practical applications.

Containment control has been a hot topic of MAS control in recent years. In [15], a hybrid control algorithm based on partial differential equations was applied to a mobile robotic network. In [16], a containment control was proposed for a first-order multi-agent system in a noisy environment. In [17] a distributed containment control algorithm was investigated for autonomous vehicles with switching communication topology. In [18], containment control of second-order MASs with time-varying delays was studied. Till now, most existing results mainly focused on containment control of homogeneous MASs, in which all agents have the same dynamics. Recently, a containment control problem was investigated for heterogeneous MASs in [19]. However, there are still few results on optimal containment control of heterogeneous MASs.

It is well known that traditional optimal control algorithms require to solve Bellman equation for discrete-time systems or Hamilton–Jacobi-Bellman (HJB) equation for continuous-time systems, whose solution is normally impossible to obtain analytically [20]. Moreover, traditional optimal control algorithms need accurate system models. As an important branch of machine learning, reinforcement learning (RL) is inspired by the fact that living creatures will modify their actions based on their interactions with the environment [22] and has its advantages in various fields. Among reinforcement learning algorithms, adaptive dynamic programming (ADP) is regarded as one of the core methodologies to solve optimal control problems. In typical ADP algorithms, two networks called actor network and critic network are utilized [23] where critic network evaluates the performance of control policies by approximating the performance indices and actor network approximates the optimal control policies. Implemented by NNs, ADP algorithms do not require analytical solutions of partial differential equations and thus have been used in various of optimal control problems including tracking control [24–26], graphical games [27–29], optimal bipartite consensus control [30] and robust control [31,32,33]. ADP algorithms were also applied to solve contain-

* Corresponding author.
E-mail address: hjp_lzu@163.com (J. Hu).

https://doi.org/10.1016/j.neucom.2020.06.106
0925-2312/

ment control problems. In [34,35], an ADP algorithm was implemented to solve the containment control problem of a MAS with unknown dynamics. In [36], an offline policy iteration based ADP algorithm was developed to solve the containment control problem of heterogeneous MAS with disturbances.

In traditional ADP architectures, only one type of reinforcement signals, the signals from environment named external reinforcement signals, were used to provide information to the critic network. However, in many practical applications which involve more complex systems, it will be necessary to obtain more informative reinforcement signals. Thus, approaches have been investigated in [37,38] in order to provide more information by modifying the external reinforcement signal, while in [39,40], a different scheme which uses another reinforcement signal to provide more information was utilized. Different from external reinforcement signals, the new reinforcement signals are generated by the controller itself and are called internal reinforcement signals. Apart from critic network and actor network, this ADP scheme contains another network called reference network, which receives information from environment and generate internal reinforcement signals. Then the internal reinforcement signals are provided to the critic network to evaluate the performance of control policies. Comparing with traditional two-network architecture, this three-network architecture can provide more information and thus facilitate the learning process [41]. Therefore, this new scheme can work more efficiently.

Motivated by the above observations and discussions, in this paper an internal reinforce adaptive dynamic programming (IR-ADP) method is proposed to solve the optimal containment control problem of continuous-time MASs. The interaction between agents is based on a communication topology and each agent only receive information from its leaders and neighbors. To speed up the learning procedure and achieve better control performance, basing on external reinforcement signals, the internal reinforcement signals are introduced and corresponding local performance indices are defined to evaluate the performance of control policies. Then an ADP algorithm with internal reinforcement signals is used to update the control policies until they reach optimum. Further, three NNs including internal reinforcement NNs, critic NNs and actor NNs are used to to implement the IR-ADP algorithm. The main contributions of this paper are given as follows: Firstly, an IR-ADP algorithm is proposed to solve the containment control problem of continuous-time MAS. A new internal reinforcement signals which contain more effective information and corresponding performance indices are designed in terms of local information. To the author's best knowledge, it is the first time that an IR-ADP algorithm is proposed to solve cooperative control problems for continuous-time MAS. Secondly, the analysis of convergence of the proposed IR-ADP algorithm is provided. It is proved that this algorithm can ensure the performance indices converge to their minimum and the control policies converge to their optimum. Thirdly, three NNs based architecture is designed to implement the proposed method. Since internal reinforcement signals are more informative, which facilitate the learning process, our proposed algorithm is more efficient. Additionally, numerical simulation results demonstrate that the proposed algorithm can make the MAS achieve containment control and the comparison between traditional ADP algorithm and our method indicates that our algorithm can facilitate the learning process and achieve the containment control with a higher convergence rate.

The rest of this paper is organized as follows: In Section 2 some preliminaries are provided. The containment control problem is formulated as well. In Section 3 the reinforcement signals and local performance indices are defined and the equivalence of the optimal control policies solved from the HJB equation and the internal reinforcement signals is analyzed. In Section 4, the PI based IR-ADP algorithm is proposed and the convergence of this algorithm is analyzed. In Section 5, the proposed algorithm is implemented by three NNs while in Section 6 some numerical simulation results are provided to demonstrate the effectiveness of this algorithm. Section 7 concludes this paper.

2. Preliminaries and problem formulation

2.1. Algebraic graph theory

Consider the communication topology between agents in a MAS as a directed graph $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathcal{A})$, where $\mathcal{V} = \{v_1, v_2, \ldots, v_n\}$ denotes a nonempty set of n vertices, $\mathcal{E} = \{(v_i, v_j) | v_i, v_i \in \mathcal{V}\} \in \mathcal{V} \times \mathcal{V}$ denotes the set of edges, and $\mathcal{A} = \{a_{ij}\}$ is the weighted adjacency matrix where $a_{ij} \geqslant 0$ are non-negative. Here, $a_{ij} > 0$ when $(v_i, v_j) \in \mathcal{E}$ and $a_{ij} = 0$ otherwise.

Let $\mathcal{N}_i = \{j | (v_i, v_j) \in \mathcal{E}\}$ be the neighbor set of node v_i, then $a_{ij} > 0$ for any $j \in \mathcal{N}_i$. Define the in-degree matrix $\mathcal{D} = diag\{d_i\}$ a diagonal matrix where $d_i = \sum_{j \in \mathcal{N}} a_{ij}$ is the weighted in-degree of node i, then the Laplacian matrix $\mathcal{L}$ can be defined as $\mathcal{L} = \mathcal{D} - \mathcal{A}$. When there exists a node v_0 which has a directed path to other nodes in the graph, then we say that this graph has a spanning tree.

2.2. Definitions and notions

Consider a MAS which has n followers and m leaders, denoted by the sets $F = \{1, 2, 3, \ldots, n\}$ and $L = \{n+1, n+2, \ldots, n+m\}$ respectively. The connection weight between the kth leader and the ith follower is defined as g_i^k. If there exists a direct connection between follower i and leader k, then $g_i^k = 1$, otherwise, $g_i^k = 0$. The weight matrix of the leader k is defined as $\mathcal{B}_k = diag\{g_i^k\}$.

Let $\mathbf{1}_n \in \mathbb{R}^n$ be the all one column vector, $I_{n\times n}$ be the n-order identity matrix and $\mathbf{0}$ be the zero matrix. Symbol $\otimes$ represents the Kronecker product and the distance from $x \in \mathbb{R}^N$ to set $\mathcal{C} \subseteq$ is denoted as $dist(x, \mathcal{C}) = \inf_{y \in \mathcal{C}} \|x - y\|_2$, where $y \in \mathcal{C}$ and $\|\cdot\|$ represents Euclidean norm. $\sigma(M)$ denotes the set of singular values of matrix M. $\bar{\sigma}(M)$ denotes the maximum singular value of M and $\underline{\sigma}(M)$ denotes the minimum singular value of M, respectively.

2.3. Problem formulation

In this paper, we consider a leader–follower MAS with n followers and m leaders. The dynamics of the ith follower can be expressed as:

$$\dot{x}_i(t) = Ax_i(t) + B_i u_i(t), \ i \in F, \tag{1}$$

where $x_i \in \mathbb{R}^N$ is the state vector and $u_i \in \mathbb{R}^{p_i}$ is the input vector, and $A \in \mathbb{R}^{N\times N}$ and $B_i \in \mathbb{R}^{N\times p_i}$ are system matrices with compatible dimensions. The dynamics of the kth leader can be expressed as:

$$\dot{x}_{0k}(t) = Ax_{0k}(t), \ k \in L, \tag{2}$$

where $x_{0k} \in \mathbb{R}^N$ is the state vector of the kth leader. In this paper, we assume that A and B_i are unknown but the pair (A, B_i) is stabilizable.

Assumption 1. The network topology $\mathcal{G}$ associated with the leader–follower system (1) and (2) is balanced (i.e., $\sum_{j \in \mathcal{N}_i} a_{ij} = \sum_{j \in \mathcal{N}_i} a_{ji}$) and every leader has a path to all the followers.

The definitions of convex set and containment control are provided as follows.

Definition 1. [42] (Convex set) A set $\mathcal{C} \subseteq \mathbb{R}^N$ is convex if for any $x, y \in \mathcal{C}$ and $\lambda \in [0,1], (1-\lambda)x + \lambda y \in \mathcal{C}$. For a finite set $X = \{x_1, x_2, \ldots, x_m\}$, its convex hull, i.e., $Co(X)$, is represented as $Co(X) = \{\sum_{i=1}^m \alpha_i x_i | x_i \in X, \alpha_i \in \mathbb{R}, \alpha_i \geqslant 0, \sum_{i=1}^m \alpha_i = 1\}$.

Definition 2. (Containment control) Define $\Delta(t) = \{Co(x_{0k}(t), k \in L)\}$. The containment control of system (1–2) achieves when all followers converge to the convex hull spanned by leaders under a given control algorithm as $t \to \infty$, i.e., for any $i \in F$:

$$\lim_{t\to\infty} \text{dist}(x_i(t), \Delta(t)) = 0.$$

The local neighborhood error vector of the ith follower now can be defined as

$$e_i(t) = \sum_{j\in\mathcal{N}_i} a_{ij}\big(x_i(t) - x_j(t)\big) + \sum_{k\in L} g_i^k (x_i(t) - x_{0k}(t)). \tag{3}$$

Let $e(t) = \big[e_1^T, e_2^T, \ldots, e_n^T\big]^T \in \mathbb{R}^{n\times N}$ and $x(t) = \big[x_1^T, x_2^T, \ldots, x_n^T\big]^T \in \mathbb{R}^{n\times N}$ denotes global error vector and global state vector respectively. Thus, (3) can be further expressed by an impact form

$$e(t) = \sum_{k=n+1}^{n+m} (\mathcal{H}_k \otimes I_N)\zeta(t),$$

where $\mathcal{H}_k = \frac{1}{m}\mathcal{L} + \mathcal{B}_k \in \mathbb{R}^{n\times n}$ and $\zeta(t) = x(t) - \bar{x}_k(t)$, $\bar{x}_k(t) = \mathbf{1}_n \otimes x_{0k}(t)$.

Remark 1. In this paper, we say that the containment control of MAS is achieved when $\lim_{t\to\infty} e(t) = 0$. According to [19], Assumption 1 can ensure that $\mathcal{H}_k^T + \mathcal{H}_k$is positive definite for the directed network with spanning trees and thus is helpful to show that $\lim_{t\to\infty} e(t) = 0$ and all the followers can converge to the convex hull spanned by the multiple leaders.

Using (1) and (3), the dynamics of the local neighborhood error for the ith agent can be written as

$$\dot{e}_i(t) = Ae_i(t) + \left(d_i + \sum_{k\in L} g_i^k\right) B_i u_i(t) - \sum_{j\in\mathcal{N}_i} a_{ij} B_j u_j(t). \tag{4}$$

It is noticed that the local neighborhood error of the ith agent is decided by the control input of the agent itself and its neighbors. Our goal in the next sections is to make $e(t) \to 0$ and thus achieve the containment control.

3. Multi-agent containment control

In order to solve the optimal containment control problems, in this section, an ADP algorithm with internal reinforcement signals is designed. The external reinforcement signal and local performance indices are defined based on the local neighborhood error (3). Besides, the internal reinforcement signal is designed and utilized to provide more information. Then the equivalence analysis shows that the optimal solutions can minimize the internal reinforcement signals and also the local performance indices.

3.1. Performance indices with internal reinforcement signals

Define the external reinforcement signal of the ith agent as:

$$r_i(e_i, u_i, u_{-i}) = \frac{1}{2}\left(e_i^T Q_{ii} e_i + u_i^T R_{ii} u_i + \sum_{j\in\mathcal{N}_i} u_j^T R_{ij} u_j\right), \tag{5}$$

where $u_{-i} = \{u_j | j \in \mathcal{N}_i\}$ is the input of the ith agent's neighbors, $Q_{ii} \in \mathbb{R}^N, R_{ii}, R_{ij} \in \mathbb{R}^p$. $Q_{ii} > 0, R_{ii} > 0$are positive definite and $R_{ij} \geqslant 0$. Then the internal reinforcement signal of the ith agent is defined as

$$s_i(e_i(t), u_i(t), u_{-i}(t)) = \int_t^\infty r_i(e_i(\tau), u_i(\tau), u_{-i}(\tau))d\tau. \tag{6}$$

From (6) it can be seen that internal reinforcement signal s_i contains the future performance of system, which can help to improve the efficiency of decision-making and thus improve the performance of control algorithm.

In order to evaluate the performance of the error dynamic system (4), the local performance index for each agent is defined as follows:

$$J_i(e_i(t), u_i(t), u_{-i}(t)) = \int_t^\infty s_i(e_i(\tau), u_i(\tau), \quad u_{-i}(\tau))d\tau. \tag{7}$$

For sake of simplicity, $s_i(e_i(t))$ and $J_i(e_i(t))$are used in sequel.

The definition of admissible control policy is provided as follows:

Definition 3. [44] (Admissible control policy) The control policy $u_i(t)$ is admissible if it can stabilize the system (4) and ensure that the local performance index (7) is finite simultaneously.

For the error dynamic system (4), the Hamilton–Jacobi-Bellman (HJB) equation is given by

$$\begin{aligned} H_i &\left(e_i, \tfrac{\partial J_i}{\partial e_i}, u_i, u_{-i}\right) \\ &\equiv \left(\tfrac{\partial J_i}{\partial s_i}\right)\left(\tfrac{\partial s_i}{\partial e_i}\right)^T \left(Ae_i + \left(d_i + \sum_{k\in L} g_i^k\right)\right. \\ &\left. \times B_i u_i - \sum_{j\in\mathcal{N}_i} a_{ij} B_j u_j\right) + s_i(e_i, u_i, u_{-i}) = 0. \end{aligned} \tag{8}$$

According to Bellman optimality principle, the optimal local performance index J_i^* satisfies

$$J_i^*(e_i(t)) = \min_{u_i(t)}\left\{\int_t^\infty s_i(e_i(\tau), u_i(\tau), u_{-i}(\tau))d\tau\right\}. \tag{9}$$

Thus, the optimal control strategy of the ith agent can be given by

$$u_i^*(t) = \arg\min_{u_i(t)}\left\{\int_t^\infty s_i(e_i(\tau), u_i(\tau), u_{-i}(\tau))d\tau\right\}. \tag{10}$$

3.2. Equivalence analysis

In this subsection, we show that the internal reinforcement signals and the local performance indices can reach their optimal values simultaneously, which means the optimal control policies can minimize the internal reinforcement signals in (6) and also the local performance indices in (7).

Lemma 1. [45] (Comparison Theorem): *If both $f(x)$ and $g(x)$ are integrable on $\mathcal{A}$ and if $g(x) \leqslant f(x)$ for every x in $\mathcal{A}$, then we have*

$$\int_{\mathcal{A}} g(x)dx \leqslant \int_{\mathcal{A}} f(x)dx.$$

The equivalence relationship between the internal reinforcement signals and the local performance indices is given by the following lemma.

Lemma 2. *If the optimal internal reinforcement signal s_i is given by:*

$$s_i^*(u_i^*, u_{-i}^*) = \min_{u_i(t)}\left\{\int_t^\infty r_i(e_i(\tau), u_i^*(\tau), u_{-i}^*(\tau))d\tau\right\},$$

then J_i and s_i reach their optimal values J_i^* and s_i^* simultaneously, that is,

$$J_i^*(u_i^*, u_{-i}^*) = \int_t^\infty s_i^*(e_i(\tau), u_i^*(\tau), u_{-i}^*(\tau))d\tau.$$

Proof. Consider a sequence of optimal controllers (u_i^*, u_{-i}^*) which satisfies

$$s_i^* \triangleq s_i(u_i^*, u_{-i}^*) \leqslant s_i(u_i, u_{-i}).$$

According to Lemma 1, we have

$$\int_t^\infty s_i(u_i^*(\tau), u_{-i}^*(\tau))d\tau \leqslant \int_t^\infty s_i(u_i(\tau), u_{-i}(\tau))d\tau,$$

which implies that

$$J_i(u_i, u_{-i}) \geqslant J_i(u_i^*, u_{-i}^*) \triangleq J_i^*. \tag{11}$$

Thus, J_i and e_i reach their optimal values J_i^* and s_i^* simultaneously, which completes the proof.

Remark 2. According to Lemma 2, since J_i^* and s_i^* reach their optimum simultaneously, thus the optimal control policy which minimizes the internal signal (6) is also an optimal solution of (9).

The Hamiltonian function of (6), which is its differential equivalence, can be written as

$$\begin{aligned} H_{si} &= \left(\frac{\partial s_i}{\partial e_i}\right)^T\left(Ae_i + \left(d_i + \sum_{k\in L} g_i^k\right)B_i u_i - \sum_{j\in\mathcal{N}_i} a_{ij}B_j u_j\right) \\ &\quad + \frac{1}{2}\left(e_i^T Q_{ii} e_i + u_i^T R_{ii} u_i + \sum_{j\in\mathcal{N}_i} u_{-i}^T R_{ij} u_{-i}\right). \end{aligned}$$

Starting from admissible control laws with boundary condition $s_i(0) = 0$. Then according to the first-order necessary condition i.e., $(\partial H_{si}^*/\partial u_i) = 0$ in optimal control theory [46], we have

$$\frac{\partial H_{si}}{\partial u_i^*} = \left(\frac{\partial s_i}{\partial e_i}\right)^T\left(d_i + \sum_{k\in L} g_i^k\right)B_i + u_i^{*T} R_{ii} = 0,$$

or equivalently,

$$u_i^{*T} R_{ii} = -\left(\frac{\partial s_i}{\partial e_i}\right)^T\left(d_i + \sum_{k\in L} g_i^k\right)B_i, \tag{12}$$

Thus the optimal control law can be determined by

$$u_i^* = -\left(d_i + \sum_{k\in L} g_i^k\right)R_{ii}^{-1}B_i^T\frac{\partial s_i}{\partial e_i}. \tag{13}$$

From (13), the optimal control law depends on the internal reinforcement signal $s_i(t)$, and thus contains a long-term information of future external reinforcement signal, which is helpful to evaluate the performance of the control law.

4. IR-ADP algorithm for optimal containment control

In fact, the HJB Eq. (8) and the optimal control policy (13) are always difficult to obtain, in this section, an iterative algorithm, i.e., IR-ADP algorithm is firstly provided to evaluate the local performance indices (7) as well as the optimal control algorithm (13). Then, the theoretical analysis is given to prove the convergence of this algorithm.

4.1. IR-ADP algorithm

We propose an IR-ADP algorithm, whose block diagram is illustrated in Fig. 1. The detailed process of the proposed algorithm is depicted in Algorithm 1.

Algorithm 1: IR-ADP algorithm

Initialization:

Let $u_i^{(0)}$, $\forall i = 1, 2, \ldots, N$ be any admissible control policy;

Iteration:

Let the iteration index $l = 0$, set the precision of computation ε;

1: **Repeat**

2: Compute the internal reinforcement signal $s_i^{(l+1)}$ by (6)

$$\left(\frac{\partial s_i^{(l+1)}}{\partial e_i}\right)^T\left(Ae_i + \left(d_i + \sum_{k\in L} g_i^k\right)B_i u_i^{(l)} - \sum_{j\in\mathcal{N}_i} a_{ij}B_j \times u_j^{(l)}\right)$$
$$+\frac{1}{2}\left(e_i^T Q_{ii} e_i + u_i^{(l)T} R_{ii} u_i^{(l)} + \sum_{j\in\mathcal{N}_i} u_{-i}^{(l)T} R_{ij} \times u_{-i}^{(l)}\right) = 0;$$

3: Compute the local performance indices $J_i^{(l+1)}$ by (7)

$$\left(\frac{\partial J_i^{(l+1)}}{\partial s_i^{(l+1)}}\right)\left(\frac{\partial s_i^{(l+1)}}{\partial e_i}\right)^T\left(Ae_i + \left(d_i + \sum_{k\in L} g_i^k\right)B_i u_i^{(l)}\right.$$
$$\left. - \sum_{j\in\mathcal{N}_i} a_{ij}B_j u_j^{(l)}\right) + s_i^{(l+1)}\left(e_i, u_i^{(l)}, u_{-i}^{(l)}\right) = 0;$$

4: Compute the optimal control policy $u_i^{(l+1)}$ by (13)

$$u_i^{(l+1)} = -\left(d_i + \sum_{k\in L} g_i^k\right)R_{ii}^{-1}B_i^T\frac{\partial s_i^{(l+1)}}{\partial e_i}; \tag{14}$$

6: **until** $\|J_i^{(l+1)} - J_i^{(l)}\| \leqslant \varepsilon$;

7: The optimal control policy and optimal local performance index can be expressed as $u_i^* = u_i^l$, $J_i^* = J_i^l$.

In the IR-ADP algorithm, let $s_i^{(l)}(t), u_i^{(l)}(t)$, and $J_i^{(l)}(t)$ be the iterative value of the internal reinforcement signal, control law and performance index, respectively. Starting from an initial admissible control policy, the algorithm firstly compute the internal reinforcement signal (6) and then local performance index (7), followed by computing the optimal control policy (13). Repeating the above processes until the difference between $J_i^{(l+1)}$ and $J_i^{(l)}$ is sufficiently small.

Remark 3. In traditional ADP algorithms, only external reinforcement signals are utilized, which are not able to provide enough information [47,48]. In our proposed IR-ADP algorithm, the internal reinforcement signals are adopted to provide more informative reinforcement signals from the controller. Moreover, introducing the internal reinforcement signals allows the controllers to evaluate the future performances, which will facilitate the learning process.

4.2. Convergence analysis of IR-ADP algorithm

In this subsection we show that s_i, J_i, u_i will converge to their optimal values s_i^*, J_i^* and u_i^*, respectively, under the proposed IR-ADP algorithm. Theorem 1 shows that s_i, J_i, u_i converge to their

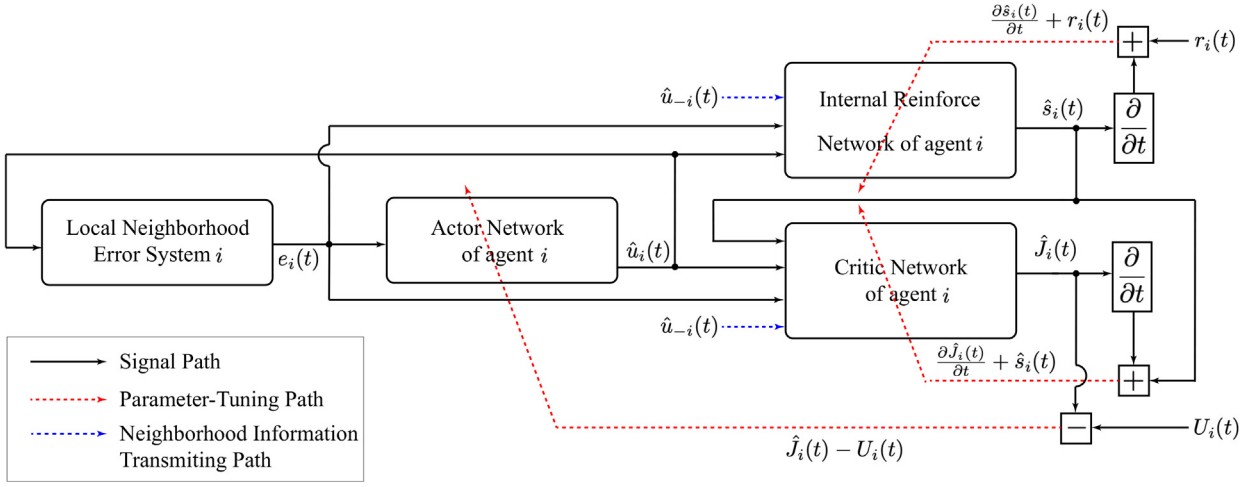


Fig. 1. Structure of IR-ADP algorithm.

optimal values when only the ith agent updates its control policy while its neighbors' control policies are fixed. Furthermore, Theorem 2 shows that s_i, J_i, u_i can converge to their optimum when all agents update their control policies simultaneously.

Theorem 1. *Assuming that only the ith agent update its control policy in every iteration while its neighbors' control policies are fixed. By performing the IR-ADP algorithm given in Algorithm 1, the iterative local performance index $J_i^{(l)}$ can converge to the optimal local performance index J_i^* given by (9).*

Proof. Consider the dynamics of the local neighborhood error

$$\dot{e}_i = Ae_i + \left(d_i + \sum_{k\in L} g_i^k\right) B_i u_i^{(l+1)} - \sum_{j\in\mathcal{N}_i} a_{ij} B_j u_j. \tag{15}$$

Since only the ith agent update its control policy in every iteration, the difference between $\dot{s}_i^{(l)}$ and $\dot{s}_i^{(l+1)}$ can be given by

$$\begin{aligned}
\dot{s}_i^{(l+1)} - \dot{s}_i^{(l)} &= \left(\frac{\partial s_i^{(l+1)}}{\partial e_i}\right)^T \left(Ae_i + \left(d_i + \sum_{k\in L} g_i^k\right) B_i u_i^{(l+1)} - \sum_{j\in\mathcal{N}_i} a_{ij} B_j u_j\right) \\
&\quad - \left(\frac{\partial s_i^{(l)}}{\partial e_i}\right)^T \left(Ae_i + \left(d_i + \sum_{k\in L} g_i^k\right) B_i u_i^{(l+1)} - \sum_{j\in\mathcal{N}} a_{ij} B_j u_j\right) \\
&= \left(\frac{\partial s_i^{(l+1)}}{\partial e_i}\right)^T \left(Ae_i + \left(d_i + \sum_{k\in L} g_i^k\right) B_i u_i^{(l+1)} - \sum_{j\in\mathcal{N}_i} a_{ij} B_j u_j\right) \\
&\quad - \left(\frac{\partial s_i^{(l)}}{\partial e_i}\right)^T \left(Ae_i + \left(d_i + \sum_{k\in L} g_i^k\right) B_i u_i^{(l)} - \sum_{j\in\mathcal{N}} a_{ij} B_j u_j\right) \\
&\quad + \left(\frac{\partial s_i^{(l)}}{\partial e_i}\right)^T \left(d_i + \sum_{k\in L} g_i^k\right) B_i u_i^{(l)} - \left(\frac{\partial s_i^{(l)}}{\partial e_i}\right)^T \left(d_i + \sum_{k\in L} g_i^k\right) \\
&\quad \times B_i u_i^{(l+1)}.
\end{aligned} \tag{16}$$

Using the fact

$$\begin{aligned}
&\left(\frac{\partial s_i}{\partial e_i}\right)^T \left(Ae_i + \left(d_i + \sum_{k\in L} g_i^k\right) B_i u_i - \sum_{j\in\mathcal{N}_i} a_{ij} B_j u_j\right) \\
&= -\frac{1}{2}\left(e_i^T Q_{ii} e_i + u_i^T R_{ii} u_i + \sum_{j\in\mathcal{N}_i} u_{-i}^T R_{ij} u_{-i}\right),
\end{aligned} \tag{17}$$

and the Eqs. (12) and (16), we have

$$\begin{aligned}
\dot{s}_i^{(l+1)} - \dot{s}_i^{(l)} &= \tfrac{1}{2} u_i^{(l)T} R_{ii} u_i^{(l)} - \tfrac{1}{2} u_i^{(l+1)T} R_{ii} u_i^{(l+1)} + \left(\frac{\partial s_i^{(l)}}{\partial e_i}\right)^T \\
&\quad \times \left(d_i + \sum_{k\in L} g_i^k\right) B_i u_i^{(l)} - \left(\frac{\partial s_i^{(l)}}{\partial e_i}\right)^T \left(d_i + \sum_{k\in L} g_i^k\right) B_i u_i^{(l+1)} \\
&= \tfrac{1}{2} u_i^{(l)T} R_{ii} u_i^{(l)} - \tfrac{1}{2} u_i^{(l+1)T} R_{ii} u_i^{(l+1)} - u_i^{(l)T} R_{ii} u_i^{(l)} + u_i^{(l)T} R_{ii} u_i^{(l+1)} \\
&= -\tfrac{1}{2} u_i^{(l)T} R_{ii} u_i^{(l)} - \tfrac{1}{2} u_i^{(l+1)T} R_{ii} u_i^{(l+1)} + u_i^{(l)T} R_{ii} u_i^{(l+1)} \\
&= -\tfrac{1}{2}\left(u_i^{(l)} - u_i^{(l+1)}\right)^T R_{ii} \left(u_i^{(l)} - u_i^{(l+1)}\right) \leqslant 0.
\end{aligned} \tag{18}$$

Then we have

$$s_i^{(l)} - s_i^{(l+1)} = -\int_t^\infty \frac{1}{2}\left(u_i^{(l+1)} - u_i^{(l)}\right)^T R_{ii} \left(u_i^{(l+1)} - u_i^{(l)}\right) d\tau \leqslant 0, \tag{19}$$

which implies that internal reinforcement signal s_i is monotonically non-decreasing. Since the admissible control policies can ensure that the local performance index J_i is finite and thus internal reinforcement signal s_i is also finite. Thus, $s_i^{(l)}$ is bounded. According to Lemma 1, then we have

$$J_i^{(l+1)} - J_i^{(l)} = \int_t^\infty \left(s_i^{(l)}(\tau) - s_i^{(l+1)}(\tau)\right) d\tau \leqslant 0, \tag{20}$$

which shows that the local performance index $J_i^{(l)}$ is monotonically non-increasing. Since the local performance indices J_i are finite and bounded as well, thus, $J_i^{(l)}$ is convergent.

Define $\lim_{l\to\infty} J_i^{(l)} = J_i^{(\infty)}$. Next, we show that $J_i^{(l)}$ converge to the optimal local performance indices J_i^*, i.e., $J_i^{(\infty)} = J_i^*$. According to the definition of J_i^* (9), we have

$$J_i^*(e_i, u_i^*, u_{-i}^*) = \min_{u_i}\{J_i(e_i, u_i, u_{-i})\} \leqslant J_i^{(\infty)}\left(e_i, u_i^{(\infty)}, u_{-i}^{(\infty)}\right).$$

Since $J_i^{(l)}$ is monotonically non-increasing, there exists an iteration index l which satisfies

$$J_i^{(\infty)}\left(e_i, u_i^{(\infty)}, u_i^{(\infty)}\right) \leqslant J_i^{(l)}\left(e_i, u_i^{(l)}, u_{-i}^{(l)}\right).$$

Let $u_i^{(l)} = u_i^*$ and $u_{-i}^{(l)} = u_{-i}^*$, then we have

$$J_i^{(\infty)}\left(e_i, u_i^{(\infty)}, u_i^{(\infty)}\right) \leqslant J_i(e_i, u_i^*, u_{-i}^*) \triangleq J_i^*.$$

Thus, we have $J_i^* = J_i^{(\infty)}$, which means that the local performance index J_i can converge to its optimum J_i^* under the proposed proposed IR-ADP algorithm. The proof is completed.

The next theorem shows that local neighborhood index J_i can converge to J_i^* when the control policies of the ith agent and its neighbors update simultaneously.

Theorem 2. *Assume that the control polices of the ith agent and its neighbors update simultaneously. Then the local performance index J_i can converge to the optimal local performance index J_i^* under the IR-ADP algorithm in Algorithm 1.*

Proof. Similar with the proof of Theorem 1, we have

$$\begin{aligned}\dot{s}_i^{(l+1)} - \dot{s}_i^{(l)} &= -\tfrac{1}{2}\left(u_i^{(l+1)} - u_i^{(l)}\right)^T R_{ii}\left(u_i^{(l+1)} - u_i^{(l)}\right) \\ &\quad -\tfrac{1}{2}\sum_{j\in\mathcal{N}}\left(u_i^{(l+1)} - u_i^{(l)}\right)^T R_{ij}\left(u_i^{(l+1)} - u_i^{(l)}\right) \\ &\quad -\sum_{j\in\mathcal{N}} u_j^{(l)T} R_{ij}\left(u_j^{(l+1)} - u_j^{(l)}\right) + \left(\frac{\partial s_i^{(l+1)}}{\partial e_i}\right)^T \\ &\quad \times\sum_{j\in\mathcal{N}} a_{ij}B_j\left(u_i^{(l+1)} - u_i^{(l)}\right).\end{aligned} \tag{21}$$

Then a sufficient condition which ensures that $\dot{s}_i^{(l+1)} - \dot{s}_i^{(l)} \leqslant 0$ is given by

$$\begin{aligned}\tfrac{1}{2}\underline{\sigma}(R_{ij})\left\|u_j^{(l+1)} - u_j^{(l)}\right\| &\geqslant \left(d_j + \sum_{k\in L} g_i^k\right)\underline{\sigma}\cdot\left(R_{jj}^{-1}R_{ij}\right) \\ &\quad \times\left\|\frac{\partial s_i^{(l)}}{\partial e_i}\right\|\|B_j\| + a_{ij}\left\|\frac{\partial s_i^{(l+1)}}{\partial e_i}\right\|\cdot\|B_j\|.\end{aligned} \tag{22}$$

If the sufficient condition in (22) holds, then we have

$$\begin{aligned}&\left(s_i^{(l+1)}(\infty) - s_i^{(l+1)}(t)\right) - \left(s_i^{(l+1)}(\infty) - s_i^{(l+1)}(t)\right) \\ =\ & s_i^{(l)} - s_i^{(l+1)} = \int_t^\infty\left(\dot{s}_i^{(l+1)} - \dot{s}_i^{(l)}\right)d\tau \leqslant 0,\end{aligned} \tag{23}$$

which implies that the internal reinforcement signal $s_i^{(l)}$ is monotonically non-decreasing. Similarly, we can show that $J_i^{(l)}$ is monotonically non-increasing and thus can converge to its optimum J_i^*.

4.3. Stability analysis

In this subsection, the stability analysis is provided for the closed-loop multi-agent system under the optimal control laws.

Theorem 3. *Consider the multi-agent system (1) with the optimal indices J_i (7) for $\forall i$. Then the closed-loop dynamics of the local error $e_i(t)$ is asymptotically stable under the proposed optimal control law (13).*

Proof. Consider the external reinforcement signal

$$r_i(e_i, u_i, u_{-i}) = \frac{1}{2}\left(e_i^T Q_{ii} e_i + u_i^T R_{ii} u_i + \sum_{j\in\mathcal{N}_i} u_j^T R_{ij} u_j\right),$$

which is positive definite. Then from Lemma 1 and (6), we have

$$s_i(e_i(t), u_i(t), u_{-i}(t)) \geqslant 0, J_i(e_i(t), u_i(t), u_{-i}(t)) \geqslant 0.$$

Since $s_i = 0$ and $J_i = 0$ only when $e_i(t) = 0$, thus s_i and J_i are positive definite. As a result, the local performance index J_i (7) can be selected as a Lyapunov function for the error dynamics.

According to the Hamiltonian in (8), we have

$$\dot{J}_i = -s_i(e_i(t), u_i(t), u_{-i}(t)) < 0,$$

which is negative definite. Therefore, the error dynamics in (4) is asymptotically stable, that is, $e_i(t) \to 0$ as $t \to \infty$, which implies that the containment control problem is solved.

5. NN implementation of IR-ADP

The IR-ADP algorithm proposed in Algorithm 1 requires an accurate system model, which is always difficult to obtain in practical applications. In this section, a data-driven implementation of the algorithm will be provided. Three NNs, i.e., internal reinforcement NN, critic NN and actor NN, are employed to estimate the internal reinforcement signal s_i, local performance index J_i and control policy u_i, respectively.

5.1. Internal reinforcement NN

The internal reinforcement neural network is used to estimate the internal reinforcement signal, which is represented as

$$\hat{s}_i(t) = W_{gi2}^T\cdot\phi\left(Z_{gi}(t)\right), \tag{24}$$

where $Z_{gi}(t) = W_{gi1}^T\cdot[e_i(t), u_i(t), u_{-i}(t)]$ is the input vector, W_{gi1} denotes the weight matrix of input-to–hidden layer,W_{gi2} denotes the weight matrix of hidden-to-output layer, and $\phi(\cdot) = tanh(\cdot)$ is the activation function.

According to (17), we define the error function e_{gi} as

$$\begin{aligned}e_{gi} &= \left(\frac{\partial s_i}{\partial e_i}\right)^T\left(Ae_i + \left(d_i + \sum_{k\in L} g_i^k\right)B_i\hat{u}_i - \sum_{j\in\mathcal{N}_i} a_{ij}B_j\hat{u}_j\right) \\ &\quad +\tfrac{1}{2}\left(e_i^T Q_{ii} e_i + \hat{u}_i^T R_{ii}\hat{u}_i + \sum_{j\in\mathcal{N}_i}\hat{u}_{-i}^T R_{ij}\hat{u}_{-i}\right),\end{aligned}$$

and the loss function of internal reinforcement network is defined as

$$E_{gi} = \frac{1}{2}e_{gi}^2.$$

For sake of convenience, during the process of training, only the hidden-to-output layer matrices W_{gi2} are updated, while the input-to-hidden layer matrices W_{gi1} are the identity matrices with appropriate dimensions. Then, according to [37], a gradient decent based weight update law is given as follows:

$$\dot{W}_{gi2} = -\beta_{gi}\cdot\frac{\partial E_{gi}}{\partial W_{gi2}} = -\beta_{gi}\frac{\partial E_{gi}}{\partial e_{gi}}\cdot\frac{\partial e_{gi}}{\partial \hat{s}_i}\cdot\frac{\partial \hat{s}_i}{\partial W_{gi2}}, \tag{25}$$

where β_{gi} is the learning rate of the internal reinforcement network.

5.2. Critic NN

The critic neural network is used to estimate the local performance index and given by

$$\widehat{J}_i(t) = W_{ci2}^T\cdot\phi(Z_{ci}(t)), \tag{26}$$

where $Z_{ci}(t) = W_{ci1}^T\cdot[\hat{s}_i(t), e_i(t), u_i(t), u_{-i}(t)]$ is the input vector represented, W_{ci1} and W_{ci2} are the weight matrices of the input-to-hidden layer and hidden-to-output layer, respectively. According to (8), the error function of critic network e_{ci} is defined as

$$\begin{aligned}e_{ci} &= H_i\left(e_i, \frac{\widehat{\partial J_i}}{\partial e_i}, \hat{u}_i, \hat{u}_{-i}\right) \\ &= \left(\frac{\widehat{\partial J_i}}{\partial \hat{s}_i}\right)\left(\frac{\partial \hat{s}_i}{\partial e_i}\right)^T\left(Ae_i + \left(d_i + \sum_{k\in L} g_i^k\right)\right. \\ &\quad \left.\times B_i\hat{u}_i - \sum_{j\in\mathcal{N}_i} a_{ij}B_j\hat{u}_j\right) + \hat{s}_i(e_i, \hat{u}_i, \hat{u}_{-i}),\end{aligned}$$

and the loss function of the critic network is defined as

$$E_{ci} = \frac{1}{2} e_{ci}^2.$$

Again, here only the weight matrices W_{ci2} are updated while W_{ci1} are the identity matrices. The gradient decent based weight update law is given by

$$\dot{W}_{ci2} = -\beta_{ci} \cdot \frac{\partial E_{gi}}{\partial W_{ci}} = -\beta_{ci} \frac{\partial E_{ci}}{\partial e_{ci}} \cdot \frac{\partial e_{ci}}{\partial \widehat{J}_i} \cdot \frac{\partial \widehat{J}_i}{\partial W_{ci2}}, \tag{27}$$

where β_{ci} is the learning rate of the critic network.

5.3. Actor NN

The actor neural network is used to generate the iterative control policy by approximating the optimal control policy (13), and given by

$$\hat{u}_i(t) = W_{ai2}^T \cdot \phi(Z_{ai}(t)), \tag{28}$$

where $Z_{ai} = W_{ai1}^T \cdot e_i$ is the input vector, W_{ai1} and W_{ai2} represent the weight matrices of the input-to–hidden layer and hidden-to-output layer, respectively. We define the error function of the actor network as

$$e_{ai} = \widehat{J}_i - U_c,$$

where U_c is the cost-to-go function and equals 0 here. Then the loss function of the actor network is defined as

$$E_{ai} = \frac{1}{2} e_{ai}^2.$$

A gradient decent based weight update law is given for W_{ai2} as follows:

$$\dot{W}_{ai2} = -\beta_{ai} \cdot \frac{\partial E_{ai}}{\partial W_{ai2}} = -\beta_{ai} \cdot \frac{\partial E_{ai}}{\partial e_{ai}} \cdot \frac{\partial e_{ai}}{\partial \hat{u}_i} \cdot \frac{\partial \hat{u}_i}{\partial W_{ai2}}, \tag{29}$$

where β_{ai} is the learning rate of the actor network.

Note that the error function of the internal reinforcement NNs and critic NNs contain the system matrices A and B_i. In order to avoid to use the knowledge of system model, we can employ an identifier to identify the system matrices. The detailed description can refer to [51,52].

The framework of the NN implementation of the proposed IR-ADP algorithm is presented in Algorithm 2.

Algorithm 2: NN implementation of IR-ADP algorithm.

Initialization:

Let $u_i^{(0)}$, $\forall i \in F$ be any admissible control policy, randomly initialize weight matrices of goal represent network, actor network and critic network $W_{gi1,2}^{(0)}, W_{ai1,2}^{(0)}, W_{ci1,2}^{(0)}$, $\forall i \in F$.

Iteration:

Let the iteration index $l = 0$, set the precision of computation ε.

1: **Repeat**

2: Compute the internal reinforcement signal $\hat{s}_i^{(l+1)}$ using (24);

3: Compute the local performance indices $\widehat{J}_i^{(l+1)}$ using (26);

4: Compute the optimal control policy $\hat{u}_i^{(l+1)}$ using (28);

5: Update the weight matrices of goal represent network using (25);

6: Update the weight matrices of critic network using (27);

7: Update the weight matrices of actor network using (29);

8: **Until** $\|\widehat{J}_i^{(l+1)} - \widehat{J}_i^{(l)}\| \leqslant \varepsilon$;

9: Return $W_{gi1,2}^{(l)}, W_{ai1,2}^{(l)}, W_{ci1,2}^{(l)}, \forall i \in F$. Then (24), (26) and (28) can be used to compute $\hat{s}_i, \widehat{J}_i$ and $\hat{u}$ respectively.

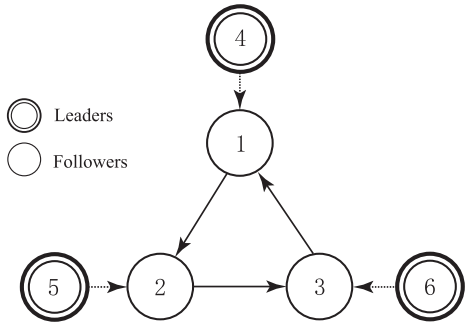
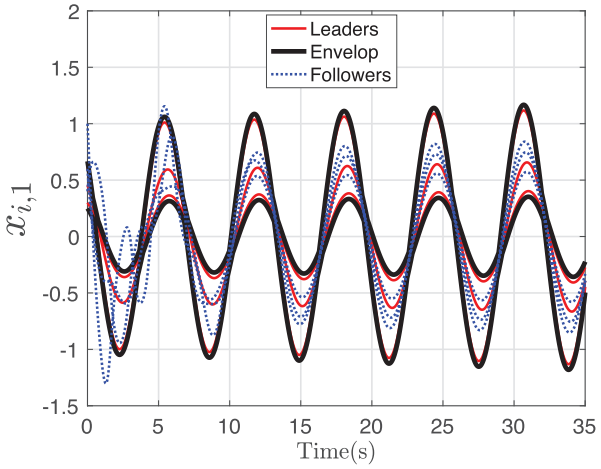
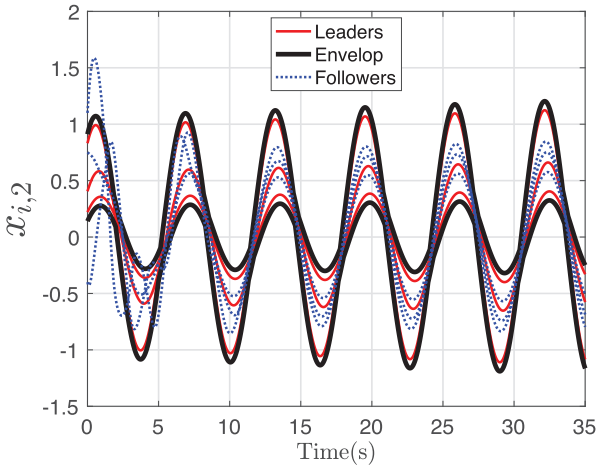


Fig. 2. The network topology for Example 1.



(a)



(b)

Fig. 3. Evolution of the agents' state $x_{i,1}$ and $x_{i,2}$.

Remark 4. Different from the traditional methods [32,26] which has a common characteristic, i.e., two-network architecture namely critic networks and actor networks are always utilized in the controller design. Different from them, the proposed IR-ADP algorithm introduce an additional internal reinforcement (IR) signal s_i. Thus, for the implementation of the proposed method,

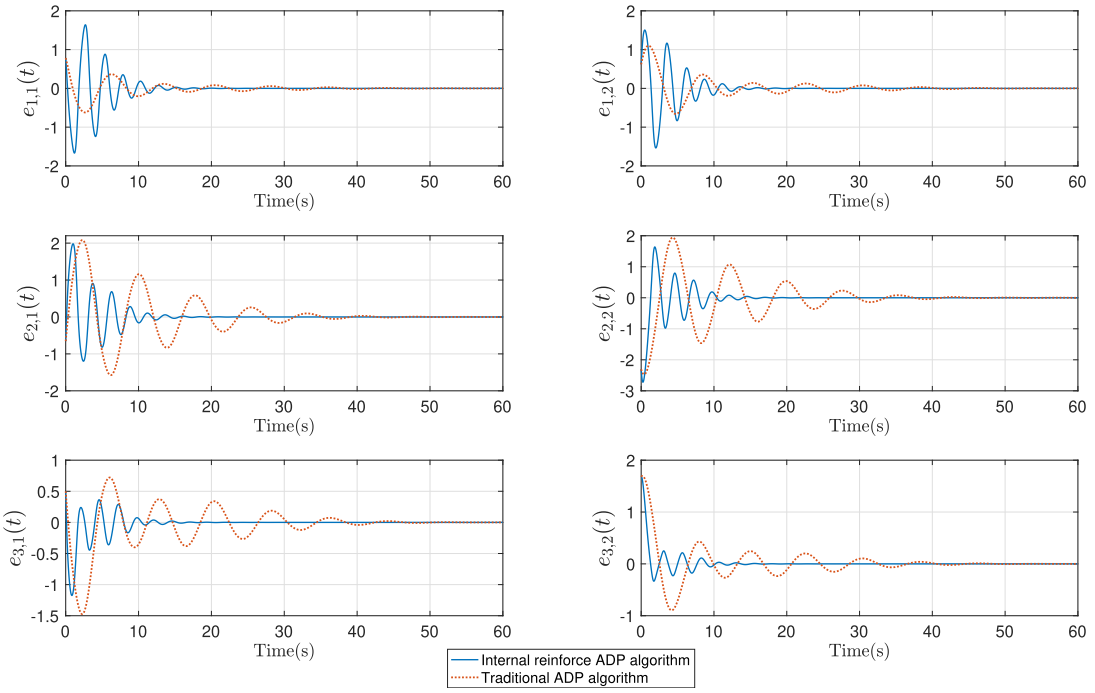


Fig. 4. Convergence comparison between our IR-ADP algorithm and traditional ADP algorithm.

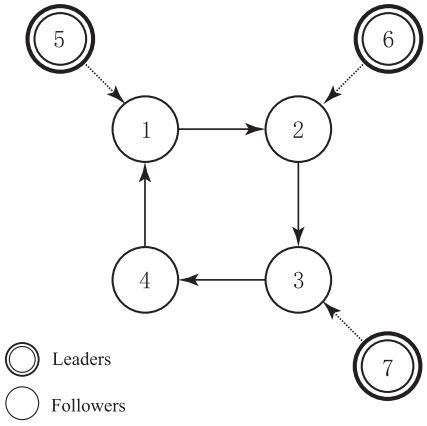
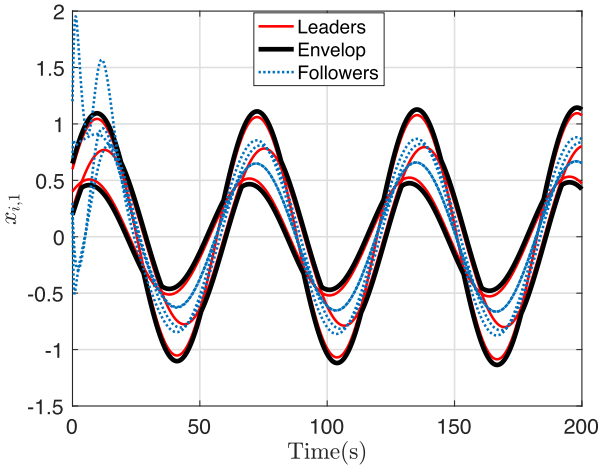
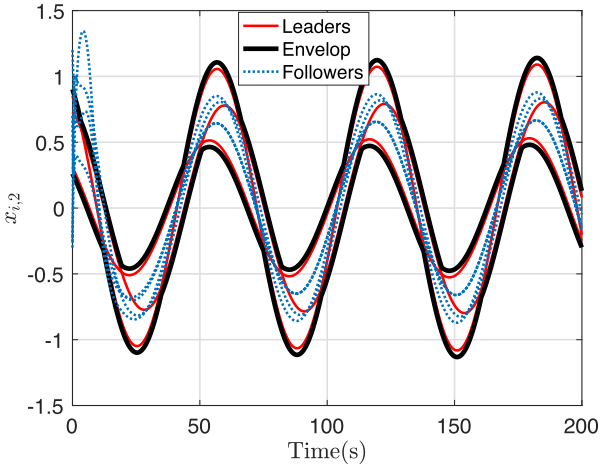


Fig. 5. The network topology for Example 2.



(a)



(b)

Fig. 6. Evolution of the agents' state $x_{i,1}$ and $x_{i,2}$.

we establishes three-networks architecture, where additional network called IR network is introduced to approximate the IR signals s_i. The advantages of our method is that the designed s_i enables agent to have more effective information in terms of local information from neighbors, which can speed up the learning procedure and achieve better control performance.

Remark 5. In the NNs implementation part, three neural networks (NNs) structure are utilized. The proposed weight update is an approximation to gradient descent. Note that in the proposed algorithm, all of the internal reinforcement NNs,actor NNs and critic NNs are updated simultaneously. The control input given by (36) is applied to system constantly, while converging to the optimal solution. Since the weight update laws for the three NNs are coupled, the critic NNs is also required to be updated constantly and simultaneously with the control input. This simultaneous update

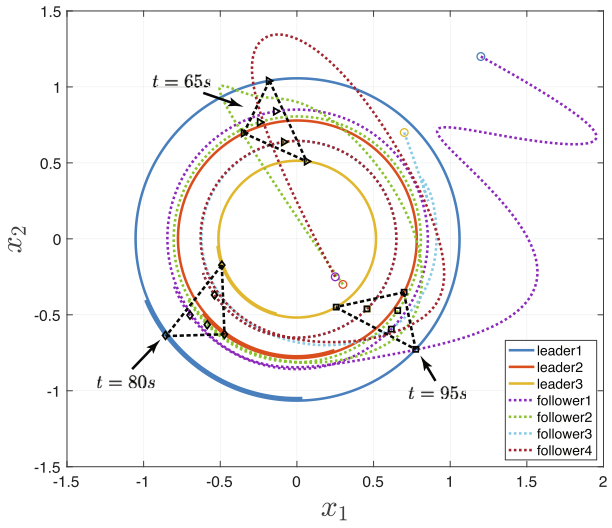


Fig. 7. Trajectory evolution of the leaders and followers.

rule for these NNs makes convergence guarantees more difficult to prove. In this instance, from the perspective of experimental studies, many researchers always depends on experience by repeating experiments to gain a better learning parameters, which proves convergence. In the works of [49,50], an exact-gradient-descent method was given with convergence guarantees.

Remark 6. It is noticed that the estimates of s_i, J_i and u_i^* are used in Algorithm 2 and thus there will certainly be some estimation errors in the implementation of the algorithm. However, according to Weierstrass higher-order approximation theorem [51], if the number of hidden layer neurons is sufficiently large, then the estimation errors can be arbitrarily small.

6. Simulation results

In this section, two numerical examples are provided to validate the proposed IR-ADP algorithm. The two examples consider the cases with different agent dynamics and network topologies.

Example 1. Consider a MAS with the follower nodes 1, 2, 3 and leader nodes 4, 5, 6. The network topology of this leader–follower system is shown in Fig. 2. From the network illustrated in Fig. 2, the interaction weights among agents are given as $a_{13} = a_{21} = a_{32} = 1$ and the weights between the leaders and followers are given as $g_1^4 = g_2^5 = g_3^6 = 1$. The matrices in the agent dynamics (1) and (2) are given by $A = \begin{bmatrix} 0.1 & -1 \\ 1 & -0.1 \end{bmatrix}$, $B_1 = \begin{bmatrix} -1.5 \\ 1 \end{bmatrix}$, $B_2 = \begin{bmatrix} -1 \\ 1 \end{bmatrix}$, and $B_3 = \begin{bmatrix} -1 \\ -0.5 \end{bmatrix}$. The weight matrices in (5) are given by $R_{11} = R_{13} = R_{21} = R_{22} = R_{32} = R_{33} = 1$, and $Q_{11} = Q_{22} = Q_{33} = I_{2\times 2}$. The initial state conditions of the leaders are given by $x_{01}(0) = \begin{bmatrix} 0.6170 \\ 0.8312 \end{bmatrix}$, $x_{02}(0) = \begin{bmatrix} 0.4550 \\ 0.4052 \end{bmatrix}$ and $x_{03}(0) = \begin{bmatrix} 0.2998 \\ 0.2206 \end{bmatrix}$. The learning rates are given by $\beta_{gi} = \beta_{ci} = \beta_{ai} = 0.005$ for $\forall i \in F = \{1, 2, 3\}$.

Fig. 3 shows the state evolution of the agents, where the red curves denote the trajectories of the leaders and the blue curves denote the trajectories of the followers. It is noted that the trajectories of the followers stay in the region formed by the envelop denoted by the black curves, which shows that the followers approach to the convex hull formed by the leaders under the proposed optimal containment control strategy.

Furthermore, Fig. 4 presents a comparison on the convergence rates of the proposed IR-ADP algorithm and the traditional ADP algorithm. From the figure, the local neighborhood errors under the proposed algorithm converge faster than those under the traditional ADP algorithm, and thus the IR-ADP algorithm has a better control performance.

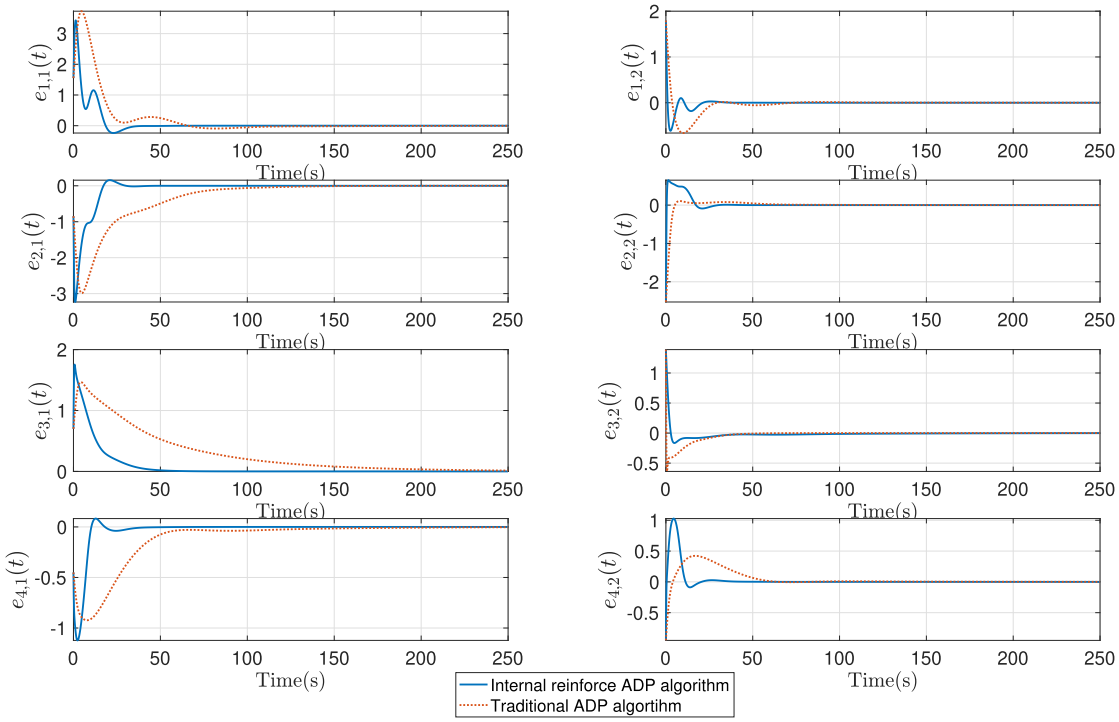


Fig. 8. Convergence comparison between our algorithm and traditional ADP algorithm in example two.

Example 2. In this example, the MAS consists of the follower nodes 1, 2, 3, 4 and leader nodes 5, 6, 7. The interaction network topology associated with the leader–follower system is illustrated in Fig. 5. For the network in Fig. 5, the interaction weights are given by $a_{14} = a_{21} = a_{32} = a_{43} = 1$ and the weights between the leaders and followers are given by $g_1^5 = g_2^6 = g_3^7 = 1$. The matrices in the agent dynamics (1) and (2) are given by $A = \begin{bmatrix} 0 & 0.1 \\ -0.1 & 0 \end{bmatrix}$, $B_1 = \begin{bmatrix} 2 \\ -1 \end{bmatrix}$, $B_2 = \begin{bmatrix} 2 \\ -3 \end{bmatrix}$, $B_3 = \begin{bmatrix} 1 \\ -3 \end{bmatrix}$, and $B_4 = \begin{bmatrix} 1 \\ -2 \end{bmatrix}$. Define the weight matrices in (5) as $R_{11} = R_{14} = R_{21} = R_{22} = R_{32} = R_{33} = R_{43} = R_{44} = 1$ while the others are 0, and $Q_{11} = Q_{22} = Q_{33} = Q_{44} = I_{2\times 2}$. Set the initial state of the leaders as $x_{01}(0) = \begin{bmatrix} 0.5990 \\ 0.8552 \end{bmatrix}$, $x_{02}(0) = \begin{bmatrix} 0.2448 \\ 0.7272 \end{bmatrix}$ and $x_{03}(0) = \begin{bmatrix} 0.4008 \\ 0.3106 \end{bmatrix}$. The learning rates are given as $\beta_{gi} = \beta_{ci} = \beta_{ai} = 0.005$ for $\forall i \in F$.

Figs. 6 and 7 show that all followers converge to the region spanned by leaders, which implies that containment control has achieved. Besides, a comparison is given in Fig. 8, which shows that the proposed IR-ADP algorithm outperforms traditional two-network ADP algorithms with faster convergence speed. Additionally, the local neighborhood errors converge after 20 s in Fig. 4 while they converge after 25 s in Fig. 8, thus the convergence speed of the proposed IR-ADP algorithm increases with the network size.

7. Conclusion

In this paper, we have investigated a containment control problem of continuous-time MASs. A data-driven IR-ADP algorithm has been applied to obtain the optimal control policies by solving HJB equation iteratively. Then three NNs have been utilized to implement the proposed algorithm, which do not need the accurate system model. The theoretical analysis has proven that the proposed algorithm can converge to the optimal control policies. Finally, the simulation results have been provided to show the effectiveness of the proposed algorithm. Future research direction includes optimal containment control of heterogeneous multi-agent systems.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

CRediT authorship contribution statement

Jiefu Zhang: Investigation, Writing - original draft. **Zhinan Peng:** Conceptualization, Methodology, Investigation. **Jiangping Hu:** Conceptualization, Methodology, Formal analysis, Writing - review & editing, Supervision. **Yiyi Zhao:** Methodology, Writing - original draft. **Bijoy Kumar Ghosh:** Writing - review & editing, Supervision.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

This work is supported partially by National Natural Science Foundation of China under Grants 61473061, 71503206, 61104104, the Sichuan Science and Technology Program under Grant 2020YFSY0012, the Fundamental Research Funds for the Central Universities under Grant JBK2002021, and the Program for New Century Excellent Talents in University under Grant NCET-13-0091.

References

[1] Y. Cao, W. Yu, W. Ren, An overview of recent progress in the study of distributed multi-agent coordination, IEEE Trans. Ind. Inf. 9 (1) (2013) 427–438.
[2] Y. Jiang, Z. Jiang, Robust adaptive dynamic programming with an application to power systems, IEEE Trans. Neural Netw. Learn. Syst. 24 (7) (2013) 1150–1156.
[3] Y. Tang, H. He, J. Wen, Power system stability control for a wind farm based on adaptive dynamic programming, IEEE Trans. Smart Grid 6 (1) (2015) 166–177.
[4] R. Olfati-Saberand, R.M. Murray, Consensus problems in networks of agents with switching topology and time-delays, IEEE Trans. Autom. Control 49 (9) (2004) 520–1533.
[5] J.J. Murray, C.J. Cox, G.G. Lendaris, Adaptive dynamic programming, IEEE Trans. Syst. Man Cybern. Part C 32 (2) (2002) 140–153.
[6] C. Mu, Z. Ni, C. Sun, Air-breathing hypersonic vehicle tracking control based on adaptive dynamic programming, IEEE Trans. Neural Netw. Learn. Syst. 28 (3) (2017) 584–598.
[7] H. Modares, I. Ranatunga, F.L. Lewis, D.O. Popa, Optimized assistive human-robot interaction using reinforcement learning, IEEE Trans. Syst. Man Cybern. 46 (3) (2016) 655–667.
[8] R. Huang, Z. Peng, H. Cheng, J. Hu, J. Qiu, C. Zou, Q. Chen, Learning-based walking assistance control strategy for a lower limb exoskeleton with hemiplegia patients, in: Proc IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 2018, pp. 2280–2285.
[9] Y. Hong, J. Hu, L. Gao, Tracking control for multi-agent consensus with an active leader and variable topology, Automatica 43 (7) (2006) 1177–1182.
[10] J. Hu, G. Feng, Distributed tracking control of leader-follower multi-agent systems under noisy measurement, Automatica 46 (8) (2010) 1382–1387.
[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] R. Olfati-Saber, J.A. Fax, R.M. Murray, Consensus and cooperation in networked multi-agent systems, Proc. IEEE 95 (1) (2007) 215–233.
[13] Y. Tian, C.L. Liu, Consensus of multi-agent systems with diverse input and communication delays, IEEE Trans. Autom. Control 53 (9) (2008) 2122–2128.
[14] 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.
[15] M. Ji, G. Ferrari-Trecate, M. Egerstedt, A. Buffa, Containment control in mobile networks, IEEE Trans. Autom. Control 53 (8) (2008) 1972–1975.
[16] J. Hu, H. Yuan, Collective coordination of multi-agent systems guided by multiple leaders, Chin. Phys. B 18 (9) (2009) 3777–3782.
[17] Y. Cao, D. Stuart, W. Ren, Z. Meng, Z, Distributed containment control for multiple autonomous vehicles with double-integrator dynamics: algorithms and experiments, IEEE Trans. Control Syst. Technol. 19 (4) (2011) 929–938.
[18] K. Liu, G. Xie, L. Wang, Containment control for second-order multi-agent systems with time-varying delays, Syst. Control Lett. 67 (2014) 24–31.
[19] H. Haghshenas, M.A. Badamchizadeh, M. Baradarannia, Containment control of heterogeneous linear multi-agent systems, Automatica 54 (2015) 210–216.
[20] R. Bellman, Dynamic programming, Science 153 (3731) (1966) 34–37.
[22] F.L. Lewis, D. Vrabie, Reinforcement learning and adaptive dynamic programming for feedback control, IEEE Circ. Syst. Mag. 9 (3) (2009) 32–50.
[23] J.J. Murray, C.J. Cox, G.G. Lendaris, R. Saeks, Adaptive dynamic programming, IEEE Trans. Syst. Man Cybern. Part C 32 (2) (2002) 140–153.
[24] 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.
[25] C. Mu, Z. Ni, C. Sun, H. He, Data-driven tracking control with adaptive dynamic programming for a class of continuous-time nonlinear systems, IEEE Trans. Cybern. 47 (6) (2016) 1460–1470.
[26] 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.
[27] K.G. Vamvoudakis, F.L. Lewis, G.R. Hudas, Multi-agent differential graphical games: online adaptive learning solution for synchronization with optimality, Automatica 48 (8) (2012) 1598–1611.
[28] 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.
[29] H. Jiang, H. Zhang, K. Zhang, X. Cui, Data-driven adaptive dynamic programming schemes for non-zero-sum games of unknown discrete-time nonlinear systems, Neurocomputing 275 (2018) 649–658.
[30] Z. Peng, J. Hu, K. Shi, R. Luo, R. Huang, B.K. Ghosh, J. Huang, A novel optimal bipartite consensus control scheme for unknown multi-agent systems via model-free reinforcement learning, Appl. Math. Comput. 369 (2020), 124812.
[31] Jiao Q., Modares H., Xu S., Lewis F.L., Vamvoudakis K.G., Disturbance rejection of multi-agent systems: a reinforcement learning differential game approach, in Proc. American Control Conference (ACC), 2015, pp. 737–742.

[32] Q. Jiao, H. Modares, S. Xu, F.L. Lewis, K.G. Vamvoudakis, Multi-agent zero-sum differential graphical games for disturbance rejection in distributed control, Automatica 69 (2016) 24–34.
[33] 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. 63 (2020), 209205.
[34] W. Wang, X. Chen, Model-free optimal containment control of multi-agent systems based on actor-critic framework, Neurocomputing 314 (2018) 242–250.
[35] Z. Peng, J. Hu, B.K. Ghosh, Data-driven containment control of discrete-time multi-agent systems via value iteration, Sci. China Inf. Sci. 63 (2020), 189205.
[36] S. Zuo, Y. Song, F.L. Lewis, A. Davoudi, Optimal robust output containment of unknown heterogeneous multiagent system using off-policy reinforcement learning, IEEE Trans. Cybern. 48 (11) (2017) 3197–3207.
[37] J. Si, Y.-T. Wang, Online learning control by association and reinforcement, IEEE Trans. Neural Netw. 12 (2) (2011) 264–276.
[38] J. Si, A.G. Barto, W. Powell, B.,Wunsch Helicopter flight control using direct neural dynamic programming, in: Handbook of Learning and Approximate Dynamic Programming, IEEE, 2004, pp. 535–559.
[39] H. He, Z. Ni, J. Fu, A three-network architecture for on-line learning and optimization based on adaptive dynamic programming, Neurocomputing 78 (1) (2012) 3–13.
[40] Z. Ni, H. He, X. Zhong, D.V. Prokhorov, Model-free dual heuristic dynamic programming, IEEE Trans. Neural Networks Learn. Syst. 26 (8) (2015) 1834–1839.
[41] X. Zhong, Z. Ni, Z.H. He, A theoretical foundation of goal representation heuristic dynamic programming, IEEE Trans. Neural Netw. Learn. Syst. 27 (12) (2015) 2513–2525.
[42] R.T. Rockafellar, Convex Analysis, Princeton University Press, New Jersey, 1972.
[44] H. Zhang, Y. Luo, D. Liu, Neural-network-based near-optimal control for a class of discrete-time affine nonlinear systems with control constraints, IEEE Trans. Neural Netw. 20 (9) (2009) 1490–1503.
[45] T.M. Apostol, One-variable Calculus, with an Introduction to Linear Algebra, vol. 1, John Wiley & Sons, New Jersey, 2007.
[46] F.L. Lewis, V.L. Syrmos, Optimal Control, John Wiley, New Jersey, 1995.
[47] F.L. Lewis, K.G. Vamvoudakis, Reinforcement learning for partially observable dynamic processes: adaptive dynamic programming using measured output data, IEEE Trans. Syst. Man Cybern. Part B 41 (1) (2010) 14–25.
[48] 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.
[49] L. Baird, Residual algorithms: Reinforcement learning with function approximation, in: Proc. 12th Int. Conf. Mach. Learn., 1995, pp. 30–37..
[50] M. Fairbank, S. Li, X. Fu, E. Alonso, D. Wunsch, An adaptive recurrent neural-network controller using a stabilization matrix and predictive inputs to solve a tracking problem under disturbances, Neural Netw. 49 (2014) 74–86.
[51] H. Modares, F.L. Lewis, M.B. Naghibi-Sistani, Adaptive optimal control of unknown constrained-input systems using policy iteration and neural networks, IEEE Trans. Neural Netw. Learn. Syst. 24 (10) (2013) 513–1525.
[52] G. Chowdhary, E. Johnson, Concurrent learning for convergence in adaptive control without persistency of excitation, 49th IEEE Conference on Decision and Control (2010) 3674–3679.

Jiefu Zhang is currently pursuing his B. Eng. degree in automation at School of Automation Engineering, University of Electronic Science and Technology of China. His current research interests include robotics, multi-agent systems and reinforcement learning.

Zhinan Peng received the B.S. degree in information and computing science from Fuyang Normal University, Fuyang, China, in 2014, and the M.S. degree in computational mathematics from the University of Electronic Science and Technology of China (UESTC), Chengdu, China, in 2016. He is currently pursuing the Ph.D. degree with the School of Automation Engineering, UESTC, Chengdu, China. His current research interests include multi-agent systems, adaptive dynamic programming, reinforcement learning.

Jiangping Hu received the B.S. degree in applied mathematics and the M.S. degree in computational mathematics from Lanzhou University, Lanzhou, China, in 2000 and 2004, respectively, and the Ph.D. degree in modelling and control of complex systems from the Academy of Mathematics and Systems Science, Chinese Academy of Sciences, Beijing, China, in 2007. He has held various positions with the Royal Institute of Technology, Stockholm, Sweden, The City University of Hong Kong, Hong Kong, Sophia University, Tokyo, Japan, and Western Sydney University, Sydney, NSW, Australia. He is currently a Professor with the School of Automation Engineering, University of Electronic Science and Technology of China, Chengdu, China. His current research interests include multi-agent systems, social dynamics, and sensor networks. Dr. Hu has served as an Associate Editor of journal Kybernetika since 2016.

Yiyi Zhao received the Master Degree in region economics from Yunnan Minzu University, Kunming, China, in 2007 and the Ph.D. degree in the School of Economics and Management, University of Electronic Science and Technology of China, Chengdu, China, in 2014. She is currently an associate professor of the School of Business Administration, Southwestern University of Finance and Economics, Chengdu, China. Her research areas include data mining and modeling and analysis of complex systems.

Rui Luo received the B.S. degree in applied mathematics from Sichuan University of Arts and Science, Dazhou, China, in 2013, and the M.S. degree in applied mathematics from the University of Electronic Science and Technology of China (UESTC), Chengdu, China, in 2017. She is currently pursuing the Ph.D. degree with the School of Automation Engineering, UESTC, Chengdu, China. Her current research interests include multi-agent systems, system identification control, neural-network-based control.

Bijoy Kumar Ghosh received the Ph.D. degree in Harvard University 1983. From 1983 to 2007, he was with the Department of Electrical and Systems Engineering, Washington University. Currently he is the Dick and Martha Brooks Regents Professor of Mathematics and Statistics at Texas Tech University, Lubbock, TX, USA. He became an IEEE Fellow in 2000, and a Fellow of the International Federation on Automatic Control in 2014. His research interests include biomechanics, cyber-physical systems and control problems in rehabilitation engineering.