Leader-follower optimal consensus of discrete-time linear multi-agent systems based on Q-learning

Ye Li[1,2], Fuyong Wang(✉)[1,2], Zhongxin Liu[1,2], and Zengqiang Chen[1,2]

[1] College of Artificial Intelligence, Nankai University, Tianjin 300350, China,
[2] Tianjin Key Laboratory of Intelligent Robotics, Nankai University, Tianjin 300350, China
wangfy@nankai.edu.cn

Abstract. This paper proposes a Q-learning solution to achieve optimal consensus of discrete-time linear leader-follower multi-agent systems. Model-free linear dynamic systems are considered herein and it is one of the topics that the control system community pays high attention to. A value function based on state error is designed for directed graph, and the weights of it can be estimated through a data-driven method. Then the consensus policies will be calculated iteratively by minimizing energy consumption. Besides, the proof of stability is given to ensure the theoretical convergence. Finally, a simple simulation is shown to verify the validity of the algorithm.

Keywords: Leader-follower consensus, optimal consensus, Q-learning, discrete-time multi-agent systems

1 Intrduction

Multi-agent systems are complex dynamic systems composed by multiple individuals which interact with each other through sensor networks. Compared with a single agent, the sophisticated, large-scale tasks can be completed by multi-agent systems with less energy consumption and simpler control policies. Thus the multi-agent technology has been applied in the fields of UAV formation control [1], satellite attitude control [2], and mobile multi-robots [3].

Multi-agent consensus is one of the basic problem of swarm intelligence, so the researches on it has grown rapidly with the development of computer and network technology. In general, leaderless model and leader-follower model are two types of consensus problems. For the leaderless model, the system achieves consensus when each agent's state achieves the same value. As for the leader-follower system, the states of all the followers should finally follow that of the leader. Xu. et al. [4] proposed a method by using the output feedback to achieve consensus under fixed and switching topologies. Then a distributed protocol was designed by Liu. et al. [5] to solve consensus problem of linear multi-agent systems over switching topologies. For the second-order system with intermittent

communication over fixed topology, a sufficient condition was given by Huang. et al. [6]. Considering the second-order Markov system under the switching topological graph, Wang. et al. [7] studied the consensus problem in the presence of network delay and state delay. Notice that all the methods mentioned above are all solutions to accurate models, while they often changes due to temperature, magnetic field, equipment aging. et al., and it is hard to acquire the precise model in many cases. Hence, we attempt to design a data-driven distributed controller for model-free systems.

Reinforcement learning is a type of machine learning methods which uses environment feedback to gain experience through constant trial and error, finally the optimal control policies can be got. Different from supervised learning, the agents don't know whether the behavior strategies are correct or not, but the value of the policies can be evaluated based on the environment feedback, and the optimal control will be realized by constantly choosing higher-value strategies ultimately. Therefore, reinforcement learning is suitable for solving complex control problems. So far, reinforcement learning has played an important role in optimal control problems, such as conditional optimal control [8], optimal control with time delays [9], optimal consensus control [10], optimal tracking control [11], adaptive dynamic programming(ADP) and zero-sum game optimal control [12]. In general, the traditional control methods are simple and efficient when models are known. In the case of unkonwn or complex models, the data-driven reinforcement learning optimal control may be a better solution.

In this paper, the model-free reinforcement learning methods are extended to multi-agent and our main contributions are shown below: 1) An optimal consensus controller is designed for leader-follower model-free systems. 2) A value function based on system state error is proposed for multi-agent systems. 3) The necessary and sufficient condition for system stability is given, which guarantees the theoretically consistent convergence.

The structure of this article is as follows. The mathematical background such as algebraic graph theory are introduced in Section 2. In Section 3, we propose a data-driven Q-learning control policies and analysis the stability of it. Then a simulation is given in Section 4. In Section 5, the work of this paper is summarized and prospected.

2 Problem Statement

2.1 Algebraic Graph Theory

The topological network of a multi-agent system is usually described as a digraph $G(V, E, A)$, where $V = (1, 2, ..., N)$ is a non-empty set with N elements representing the agents in the system and $E \subseteq (V \times V)$ stands for the edge set. The matrix $A = [a_{ij}]$ is defined as the adjacency matrix whose element $a_{ij} \geq 0$ representing the connection weight between nodes v_i and v_j, when the information flows from v_j to v_i, $a_{ij} > 0$, otherwise $a_{ij} = 0$, and $a_{ii} = 0$. Define the in-degree of each agent as $d_i = \sum_{j=1}^{n} a_{ij}$, then the in-degree of the graph can be expressed

as $D = diag\{d_1, ...d_n\}$. The Laplacian matrix, expressed as $L = D - A$, shows all the communication information of the digraph. The neighbor information of each agent is stored in the set $N_i = \{j|v_j : (v_j, v_i) \in E, a_{ij} \neq 0\}$. A graph is called strongly connected graph if a directed path can reach to node j form any node i at least. Such a graph who has a root node which can reach any other node through a certain path is called owning a directed spanning tree. The connection weight between the leader and followers is defined as $\mathcal{G} = \{g_1, g_2, ..., g_n\}$, if the follower can interact with the leader, $g_i > 0$, otherwise $g_i = 0$.

2.2 Optimal Consensus Control of Multi-agent Systems

For a discrete-time multi-agent system with N followers, the dynamic equation of each agent is as follows:

$$x_i(k+1) = Ax_i(k) + Bu_i(k), \;\; i = 1, 2, ..., N. \tag{1}$$

where $x_i \in R^n$ is the state of agent i, and $u_i \in R^n$ is the control policy of the system, A and B are the state matrixes which are considered as unkonwn. Note that A and B are the same for all angets.

The agent responsible for generating the trajectory is defined as the leader whose dynamic equation is:

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

where $x_0 \in R^n$ is the state of the leader.

Remark 1. It is of little significance to the problem of multi-agent consensus if the trajectory converges to zero, so the leader should generate concussive or divergent paths, which means that the eigenvalues of matrix A should be on or outside the unit circle.

The goal of this paper is to design distributed control policies which can drive all the followers converging to the state of the leader, in other words, when $\lim_{k\to\infty} ||x_i(k) - x_0(k)|| = 0, \forall i$, the consensus state is reached. So the local tracking error of each agent is:

$$\varepsilon_i(k) = \sum_{j \in N_i} a_{ij}(x_j(k) - x_i(k)) + b_i(x_0(k) - x_i(k)) \tag{3}$$

where a_{ij} is the communication weight between followers, and b_i is the weight between the leader and followers, if agent i can interact with the leader, $b_i > 0$, otherwise $b_i = 0$.

Define the global tracking error vector as:

$$\varepsilon(k) = [{\varepsilon_1}^T, {\varepsilon_2}^T, ..., {\varepsilon_N}^T]^T \in R^{nN} \tag{4}$$

Then Equation (3) can be rewitten as:

$$\begin{aligned} \varepsilon(k) &= (L \otimes I_n)\overline{x}_0(k) - (L \otimes I_n)x(k) + (\mathcal{B} \otimes I_n)(\overline{x}_0(k) - x(k)) \\ &= -((L + \mathcal{B}) \otimes I_n)(x(k) - \overline{x}_0(k)) \end{aligned} \tag{5}$$

where L is the Laplacian matrix, $B = [b_{ij}] \in R^{N\times N}$ is the connection weight matrix between the leader and followers with $b_{ii} = b_i$, $x(k) = [{x_1}^T(k), {x_2}^T(k), ..., {x_n}^T(k)]^T \in R^{nN}$ is the global state vector, $\overline{x}_0(k) = [{x_0}^T, {x_0}^T, ..., {x_0}^T]^T \in R^{nN}$ is designed to calculate the state error of each follower and leader.

There is another type of error called states error which can be defined as follows:

$$\eta(k) = x(k) - \overline{x}_0(k) \tag{6}$$

Then equation (5) can be rewritten as:

$$\varepsilon(k) = -((L + \mathcal{B}) \otimes I_n)\eta(k) \tag{7}$$

Lemma 1. *[13] If matrix $(L + \mathcal{B})$ is non-singular, then the system state error is bounded:*

$$||\eta(k)|| \leq (\lambda_{\min}(L + \mathcal{B})^{-1})||\varepsilon(k)|| \tag{8}$$

So we only need to guarantee $\varepsilon(k)$ is convergent then the system can achieve consensus according to equation (8).

The following assumptions are necessary to ensure the existence of a stable controller u can be calculated finally:

Assumption 1 *State matrix (A,B) are cntrollabel and unknown.*

Assumption 2 *The digraph G owns a spanning tree and $g_i \neq 0$ for a root node at least.*

2.3 The Definition of Value Function(Q function)

In order to push the state of followers coverge to the leader, a suitable value function is significant. The designing Q function is as follows:

$$V(x_k) = \sum_{i=k}^{\infty} ({\varepsilon_i}^T Q \varepsilon_i + {u_i}^T R u_i) = \sum_{i=k}^{\infty} r_i \tag{9}$$

with weighting matrix $Q = Q^T \geq 0$ and $R = R^T > 0$. For each iteration:

$$r_i = {\varepsilon_i}^T Q \varepsilon_i + {u_i}^T R u_i \tag{10}$$

Both the optimal control policy ${u_k}^*$ and the minimum energy consumption ${V_k}^*$ will be obtained when system achieved consensus:

$$V^*(x_k) = \min_u \sum_{i=k}^{\infty} ({\varepsilon_i}^T Q \varepsilon_i + {u_i}^T R u_i) \tag{11}$$

Lemma 2. *Admissible Control [14]: A control policy $u_i(k), \forall i$ is called to be adimssible if the policy can not only stabilze the system, but also ensure that the Q-function is finite.*

In order to facilitate iterative calculation, the Q-function can be written in the form of the Bellman equation under the premise of admissible control:

$$\begin{aligned} V(x_k) &= ({\varepsilon_k}^T Q \varepsilon_k + {u_k}^T R u_k) + \gamma \sum_{i=k+1}^{\infty} ({\varepsilon_i}^T Q \varepsilon_i + {u_i}^T R u_i) \\ &= r_k + \gamma V(x_{k+1}) \end{aligned} \tag{12}$$

where $0 < \gamma \leq 1$ is the discount factor.

Lemma 3. *The Q-function can be written in quadratic form for the Linear Quadratic Regulator(LQR) cases:*

$$V(x_k) = {x_k}^T P x_k \tag{13}$$

for some $n \times n$ matrix P.

Thus the value function can be rewritten as:

$${x_k}^T P x_k = {\varepsilon_k}^T Q \varepsilon_k + {u_k}^T R u_k + \gamma {x^{\mathrm{T}}}_{k+1} P x_{k+1} \tag{14}$$

The value function of agent i satisfies the discrete HJB equation:

$$V^*(x_k) = \min_{u_i(k)} \{({\varepsilon_k}^T Q \varepsilon_k + {u_k}^T R u_k) + \gamma V^*(x_{k+1})\} \tag{15}$$

The optimal control policies can be obtained by seeking the partial derivative $\partial V^*(x_k)/\partial u_k = 0$:

$${u^*}_i(k) = \arg\min_{u_i(k)} \{({\varepsilon_k}^T Q \varepsilon_k + {u_k}^T R u_k) + \gamma V^*(x_{k+1})\} \tag{16}$$

Lemma 4. *Golbal Nash Equilibrium Solution [14]: If the control sequence including N control policies is the golbal Nash equilibrium solution of the game, then the following formula holds:*

$$\begin{aligned} V_i^*(k) &= V_i({u_1}^*, {u_2}^*, ..., {u_i}^*, ..., {u_N}^*) \\ &\leq V_i({u_1}^*, {u_2}^*, ..., u_i, ..., {u_N}^*) \end{aligned} \tag{17}$$

According to Lemma 3, the Q-function of agent i is:

$$V^*(x_k) = \min_{u_i(k)} \{({\varepsilon_k}^T Q \varepsilon_k + {u^*}_k^T R {u^*}_k) + \gamma V^*(x_{k+1})\} \tag{18}$$

3 Multi-agent Consensus Algorithm Based on Q-learning

3.1 Q-learning Algorithm

Due to the Q-function can be expressed in a quadratic form, the value function was constructed as follows:

$$\begin{aligned} Q_i(\varepsilon_k, u_k) &= \begin{bmatrix} \overline{\varepsilon}_i(k) \\ u_i(k) \end{bmatrix}^T H_i \begin{bmatrix} \overline{\varepsilon}_i(k) \\ u_i(k) \end{bmatrix} \\ &= {z_i}^T(k) H_i z_i(k) \end{aligned} \tag{19}$$

where $\overline{\varepsilon}_i(k) = [{\varepsilon_i}^T(k), {\varepsilon_{ij1}}^T(k), {\varepsilon_{ij2}}^T(k), ..., {\varepsilon_{ijp}}^T(k)]^T \in R^{n(p+1)}$, $z_i(k) = [{\overline{\varepsilon}_i}^T(k), {u_i}^T(k)]^T \in R^{n(p+1)+m}$, $H_i = {H_i}^T$, $j1, j2, ..., jp \in N_i$, And H_i is the weight matrix of the Q-function:

$$\begin{aligned} H_i &= \begin{bmatrix} H_{\varepsilon\varepsilon} & H_{\varepsilon u_i} \\ H_{u_i\varepsilon} & H_{u_i u_i} \end{bmatrix} \\ &= \begin{bmatrix} H_{i(\varepsilon_i,\varepsilon_i)} & H_{i(\varepsilon_i,\varepsilon_{j1})} & \cdots & H_{i(\varepsilon_i,\varepsilon_{jp})} & H_{i(\varepsilon_i,u_i)} \\ H_{i(\varepsilon_{j1},\varepsilon_i)} & H_{i(\varepsilon_{j1},\varepsilon_{j1})} & \cdots & H_{i(\varepsilon_{j1},\varepsilon_{jp})} & H_{i(\varepsilon_{j1},u_i)} \\ \vdots & \vdots & & \vdots & \vdots \\ H_{i(\varepsilon_{jp},\varepsilon_i)} & H_{i(\varepsilon_{jp},\varepsilon_{j1})} & \cdots & H_{i(\varepsilon_{jp},\varepsilon_{jp})} & H_{i(\varepsilon_{jp},u_i)} \\ H_{i(u_i,\varepsilon_i)} & H_{i(u_i,\varepsilon_{j1})} & \cdots & H_{i(u_i,\varepsilon_{jp})} & H_{i(u_i,u_i)} \end{bmatrix} \end{aligned} \tag{20}$$

where $j_1, j_2, ..., j_p \in N_i$ are the number of neighbors of the agent i over the topology G. The $H_{i(u_i(k),u_i(k))}, H_{i(\varepsilon_i(k),u_i(k))}$ are expressed into the form $H_{i(u_i,u_i)}, H_{i(\varepsilon_i,u_i)}$ for a more concise equation.

Then find the partial derivative with respect to $u_i(k)$ to obtain the optimal control policies and minimize the Q-function:

$$\frac{\partial Q_i}{\partial u_i(k)} = 2H_{i(u_i,u_i)}u_i(k) + 2H_{i(\varepsilon_i,u_i)}\varepsilon_i(k) + \sum_{j\in N_i} H_{i(u_i,\varepsilon_j)}\varepsilon_j(k) = 0 \tag{21}$$

The optimal control can be received by solving the euqation (21):

$$\begin{aligned} u_i(k) &= -H^{-1}(H_{i(u_i,\varepsilon i)}\varepsilon_i(k) + \sum_{j\in N_i} H_{i(u_i,\varepsilon_j)}\varepsilon_j(k)) \\ &= -H^{-1}\widetilde{H}_i\varepsilon(k) = K_i\varepsilon(k) \end{aligned} \tag{22}$$

where $\widetilde{H}_i = [H_{i(u_i,\varepsilon_i)}, H_{i(u_i,\varepsilon_{j1})}, ..., H_{i(u_i,\varepsilon_{jp})}]$, $K_i = H^{-1}\widetilde{H}_i$ is the feedback control matrix.

Any model information is not needed during the strategy iteration, and the weight matrix H can be solved by the Least Square(LS) equation.

$$\begin{aligned} Q_i(\varepsilon_k, u_k) &= {z_i}^T(k)H_i z_i(k) \\ &= v(z_i(k){z_i}^T(k))v(H_i) \\ &= \overline{Z}_i(k)h_i \end{aligned} \tag{23}$$

where the function $v(.)$ is to stack an $m \times m$ matrix into an $m(m+1)/2$ column vector. The rule is stacking m column vectors starting with the diagonal element and ending with the last element in each column into a vector by column index.

If $Q_i(\varepsilon_k, u_k)$ has full column rank, then h_i can be directly solved through LS method:

$$h_i = (\overline{z}_i(k){\overline{z}_i}^T(k))\overline{z}_i(k)Q_i(\varepsilon_k, u_k) \tag{24}$$

The complete algorithm flow is shown in Algorithm 1.

3.2 Stablity Analysis

Theorem 1. *Consider the leader-follower multi-agent system (1), (2), if Assumptions 1-2 hold, then the optimal consensus problem can be solved through the Q-learning method (22) and the energy consumption (19) will be minimized.*

Algorithm 1. Q-learning algorithm for multiagent system

Step 1, Initialization: Choose admissible control policies ${u_i}^0$ for each agent i. Let j denote the iteration index, and N denotes the the max training iteration.
Step 2, Policy Evaluation: Solve the weight matrix H:
$h_i = (\overline{z}_i(k){\overline{z}_i}^T(k))\overline{z}_i(k)Q_i(\varepsilon_k, u_k)$
Step 3, Policy Update: Calculating the control policies according to matrix H:
${u_i}^{j+1}(k) = -H^{-1}(H_{i(u_i,\varepsilon i)}\varepsilon_i(k) + \sum_{j\in N_i} H_{i(u_i,\varepsilon_j)}\varepsilon_j(k))$
step 4, Finish : Stop when $j = N$, otherwise set $j = j + 1$, $k = k + 1$ and jump to the step 2.

Proof. According to equation(12), the single error obtained by the system is:

$$r_k = V(x_k) - \gamma V(x_{k+1})$$

Define the Lyapunov difference function is:

$$\Delta(\gamma^k V(x_{k+1})) = \gamma^{k+1}V(x_{k+1}) - \gamma^k V(x_{k+1})$$

Then the difference fuction can be rewritten as:

$$\Delta(\gamma^k V(x_{k+1})) = -\gamma r_k \leq 0$$

The above formulas show that when $k \to \infty$, $\varepsilon_i(k) \to 0$, so the system (1), (2) will asymptotically stable and the state error $\eta(k)$ will tend to 0 eventually.

4 System Simulation

In this section, a simple simulation is given to verify the effectiveness of the algorithm. Suppose a model-free multi-agent system are formed by a leader and three followers, the information flows undirectionally in the topology G. Fig.1 shows the topology of the system.

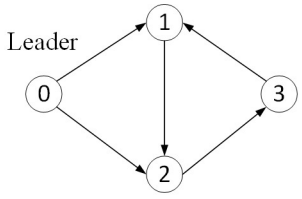

Fig. 1. System topology

The system matrix is given to simulate the dynamic environment:

$$A = \begin{bmatrix} 0.995 & -0.194 \\ 0.194 & 0.995 \end{bmatrix}, \quad B = \begin{bmatrix} 0.8 & -1 \\ 0 & 0.9 \end{bmatrix}$$

The initial state of the system is:

$$X_0{}^0 = \begin{bmatrix} 5 \\ -3 \end{bmatrix}, \quad X_1{}^0 = \begin{bmatrix} 12 \\ 26 \end{bmatrix}, \quad X_2{}^0 = \begin{bmatrix} -18 \\ 9 \end{bmatrix}, \quad X_3{}^0 = \begin{bmatrix} 24 \\ -23 \end{bmatrix}$$

where X_0 is the state of the leader, X_1, X_2, X_3 are the states of the followers. The weight matrix H is initialized as follows:

$$H_i{}^0 = \begin{bmatrix} 0.1 & 0 & 0 & 0 & 0.1 & 0.1 \\ 0 & 0.1 & 0 & 0 & 0.1 & 0.1 \\ 0 & 0 & 0.1 & 0 & 0.1 & 0.1 \\ 0 & 0 & 0 & 0.1 & 0.1 & 0.1 \\ 0.1 & 0.1 & 0.1 & 0.1 & 1 & 0 \\ 0.1 & 0.1 & 0.1 & 0.1 & 0 & 1 \end{bmatrix}$$

The corresponding initial admissible control policies are:

$$u_1{}^0 = \begin{bmatrix} 0.75 \\ 0.75 \end{bmatrix}, \quad u_2{}^0 = \begin{bmatrix} -2.4 \\ -2.4 \end{bmatrix}, \quad u_3{}^0 = \begin{bmatrix} 4.15 \\ 4.15 \end{bmatrix}$$

The discont factor is 0.8 and the values of matrix Q and R are as follows:

$$Q = \begin{bmatrix} 3 & 0 \\ 0 & 2 \end{bmatrix}, \qquad R = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

The variation of the system error with the number of iterations is shown in Fig.2, the consensus error incereased in the early 10 steps because of the lack of system information. Then the error gradually decreases and reaches a stable value around 100 steps with the iteration progressing.

The states tracking are shown in Fig.3, they are significantly different around 40 steps and the fluctuations of each individual are large due to the adjusting of the controller. The error between followers and leader as well as the fluctuation of the follower's state are gradually decreasing as time goes by, after 100 iterations, they all reach consensus.

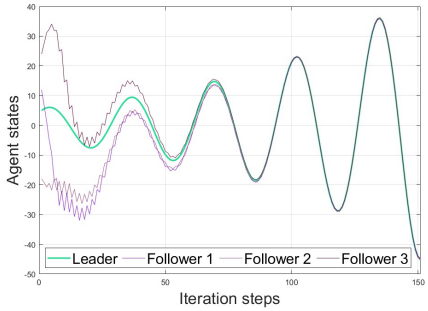

Fig. 2. Diagram of the tracking states

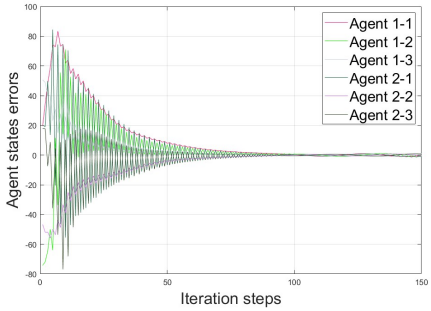

Fig. 3. Diagram of the consensus errors

The energy consumption changes are shown in Fig.4. As the figure shows that the energy consumption increases due to the model is unknown within 20 steps and dropped rapidly at 20-50 steps. After about 70 steps, the energy consumption stablizes and maintains at a low level. So the proposed algorithm can achieve optimal consensus while minimizing energy consumption.

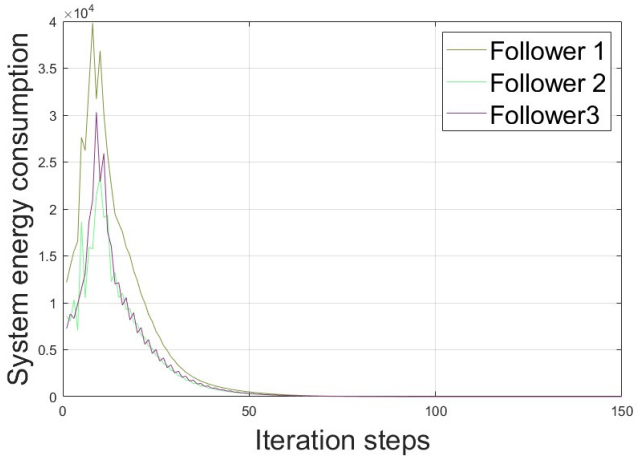

Fig. 4. Diagram of the energy consumption

5 Conclusion

In this paper, a class of discrete-time model-free multi-agent system is studied. In order to avoid complex systematic modeling, we proposed a algorithm based on Q-learning. Through this method, the system can not only achieve consensus, but also minimize the energy consumption. The stability analysis is also given. Finally, a simple simulation is shown to verify the effectiveness of the metioned method. In the future, the continuous mulit-agent system consensus problem will be considered with Deep-Q-learning method.

6 Acknowledgements

This work is supported by the Tianjin Natural Science Foundation of China (Grant Nos. 20JCQNJC01450, 20JCYBJC01060), the National Natural Science Foundation of China (Grant No. 61973175), and the Fundamental Research Funds for the Central Universities, Nankai University (No. 63211119).

References

1. Lai Y H, Li R, Shi J Y, et al. On the study of a multi-quadrotor formation control with triangular structure based on Graph theory[J]. Control Theory Appl, 2018, 35: 1530-1537.
2. Meng Z, Ren W, You Z. Distributed finite-time attitude containment control for multiple rigid bodies[J]. Automatica, 2010, 46(12): 2092-2099.
3. Jiang Y T, Liu Z X, Chen Z Q. Distributed finitetime consensus algorithm for multiple nonholonomic mobile robots with disturbances[J]. Control Theory and Applications, 2014, 31(04): 531-537.
4. Xu J, Xie L, Li T, et al. Consensus of multi-agent systems with general linear dynamics via dynamic output feedback control[J]. IET Control Theory and Applications, 2013, 7(1): 108-115.
5. Liu X K, Wang Y W, Xiao J W, et al. Distributed hierarchical control design of coupled heterogeneous linear systems under switching networks[J]. International Journal of Robust and Nonlinear Control, 2017, 27(8): 1242-1259.
6. Huang N, Duan Z, Zhao Y. Leader-following consensus of second-order non-linear multi-agent systems with directed intermittent communication[J]. IET Control Theory and Applications, 2014, 8(10): 782-795.
7. Yi J W, Wang Y W, Xiao J W, et al. Consensus in second-order Markovian jump multi-agent systems via impulsive control using sampled information with heterogenous delays[J]. Asian Journal of Control, 2016, 18(5): 1940-1949.
8. Modares H, Lewis F L, Naghibi-Sistani M B. Adaptive optimal control of unknown constrained-input systems using policy iteration and neural networks[J]. IEEE Transactions on Neural Networks and Learning Systems, 2013, 24(10): 1513-1525.
9. Wang B, Zhao D, Alippi C, et al. Dual heuristic dynamic programming for nonlinear discrete-time uncertain systems with state delay[J]. Neurocomputing, 2014, 134: 222-229.
10. Olfati-Saber R, Murray R M. Consensus problems in networks of agents with switching topology and time-delays[J]. IEEE Transactions on Automatic Control, 2004, 49(9): 1520-1533.
11. Jiang Y, Fan J, Chai T, et al. Tracking control for linear discrete-time networked control systems with unknown dynamics and dropout[J]. IEEE Transactions on Neural Networks and Learning Systems, 2017, 29(10): 4607-4620.
12. Al-Tamimi A, Lewis F L, Abu-Khalaf M. Model-free Q-learning designs for linear discrete-time zero-sum games with application to H-infinity control[J]. Automatica, 2007, 43(3): 473-481.
13. Abouheaf M I, Lewis F L, Vamvoudakis K G, et al. Multi-agent discrete-time graphical games and reinforcement learning solutions[J]. Automatica, 2014, 50(12): 3038-3053.
14. Zhang H, Jiang H, Luo Y, et al. Data-driven optimal consensus control for discrete-time multi-agent systems with unknown dynamics using reinforcement learning method[J]. IEEE Transactions on Industrial Electronics, 2016, 64(5): 4091-4100.