A model-free synchronization solution for linear discrete-time multi-agent systems based on A3C algorithm

Ye Li, Zhongxin Liu*, Zengqiang Chen

College of Artificial Intelligence, Nankai University, Tianjin 300350, China
*E-mail: lzhx@nankai.edu.cn

This paper proposes a synchronization solution for model-free discrete-time leader-following systems based on the Asynchronous Advantage Actor-Critic (A3C) algorithm. The optimization object is a value function constructed by the consensus error. Furthermore, the multi-concurrency training method is applied to train the act net and the critic net, which are the nets responsible for generating optimal policies and estimating the value of the error-action pair. In this way, time-related data of the system is turned into independent and identically distributed data, ensuring the feasibility and speed of the algorithm. Finally, a simple simulation is provided to validate the efficiency of the proposed solution.

Keywords: Multi-agent Systems; Synchronization; Reinforcement Learning; Asynchronous Advantage Actor-Critic algorithm (A3C)

1. Introduction

With the development of computer science, multi-agent systems (MASs) have received more and more attention. Compared to a single agent, the MASs can work jointly and have better environmental awareness, allowing it to complete more complex tasks. In recent years, the MASs technology has been used in various applications such as drone formation[1], intelligent transportation[1], and robotics[2].

Since the synchronization problem is one of the basic problems for MASs technologies, the related issues have been widely studied. It can be divided into leaderless MASs synchronization problem and the leader-following MASs synchronization problem, the former requires the agents eventually reach the same uncertain state, and the latter requires the states of the following agents be the same with the leader. Considering various possible situations in real applications, a lot of work such as fault-tolerant control[3], synchronization problems for heterogeneous situations, formation control, etc. are also carried out. With the deepening of the research on this

problem, more performances indicators need to be satisfied, and this is the optimal synchronization problem. From continuous-time MASs to discrete-time MASs[4], from leader-following MASs to leaderless MASs[3], and from homogeneous MASs to heterogeneous MASs[5], a variety of results have been obtained. However, all the above results are model-based methods, when the models are unknown, we have to estimate the models or design complex observers, and these jobs are complex and time-consuming.

Reinforcement learning (RL) is a kind of methods that can gain experience and achieve optimal control by interacting with the environment. Different from supervised learning methods, the RL methods don't need to prepare datasets with labels in advance, so it is a good way to solve the optimal control problems. In 2009, the RL methods were used to calculate the controllers for continuous-time systems[6]. In 2014, the integral reinforcement learning (IRL) method is proposed to solve optimal control problems for linear or nonlinear systems with partially unknown models[7]. Recently, Li. et al. proposed a Q-learning based method to generate the synchronization controller for model-free discrete-time leader-following MASs[8]. However, the above methods can't solve the dimensional explosion problem when the number of agents increases.

In this paper, a novel model-free synchronization solution based on A3C is proposed, and the main contributions are as follows: 1) The dimension extension problem has been solved. The action space and the error-action space are all fitted by neural networks, which avoid high-dimensional matrix calculations and time-consuming policy search processes. 2) The parallel training method is used. In this way, the time-related data generated by MASs are scrambled, and turns into independent and identically distributed data, while speeding up the training process. 3) The proposed method is highly flexible and strongly robust. Many parameters can be adjusted to meet different performances requirements such as control accuracy, synchronization time, etc. Besides, when the system properties change due to temperature, pressure, aging, etc. the controller can adjust the control strategy in real time to achieve optimal control.

The structure of this paper is as follows: In Section 2, some basic knowledge about MASs is introduced, then in Section 3, the A3C based controller is constructed. In Section 4, a simple simulation example is given to show the effectiveness of the proposed solution.

Notations: In this paper, matrix I_n is the $n \times n$ dimensional identity matrix. $\delta_{\min}(\cdot)$ is the function to find the minimum eigenvalue of a matrix. The $L_2 - norm$ of the vector s is defined as $||s||$.

2. Problem Statement

2.1. Algebraic Graph Theory

The topological network for a discrete-time leader-following MASs can be represented as a diagraph $G = (V, E, A)$, where $V = \{v_1, v_2, \cdots, v_n\}$ is the node set with N agents, $E = \{(i, j) \in V \times V\}$ represented the edge set, and $\mathbb{N} = \{1, 2, 3, \cdots, N\}$ is the subscript set. $A = \{a_{ij} \in R^{n\times n} \mid i, j \in \mathbb{N}\}$ is the non-negative adjacency matrix, which contains the communication information between agents. If agent i can receive information from agent j, then $a_{ij} > 0$, otherwise $a_{ij} = 0$. Note that the topologies considered in this paper are simple graphs, so $a_{ii} = 0,\ \forall i, j \in \mathbb{N}$. Then the neighbor information can be represented as $N_i = \{j \mid v_j : (v_j, v_i) \in E, a_{ij} \neq 0\}$. The in-degree of each agent i can be defined as $d_{in}(v_i) = \sum_{j=1}^{n} a_{ij}$, and the in-degree matrix of the system is $D = diag\{d_{in}(v_i) \mid i = 1, 2, 3, \cdots, n\}$. The laplace matrix of the system is defined as $L = D - A$. The connecting matrix between following agents and leader agent is defined as $F = diag\{f_i \mid i = 1, 2, 3, \cdots, n\}$, if the following agent can receive the information from the leader agent i, then $f_i > 0$, otherwise $f_i = 0$.

2.2. Optimal Synchronization Problem for Leader-following System

Considering a discrete-time leader-following MASs with $N + 1$ agents, the dynamical equations are as follows:

$$\begin{cases} x_i(k+1) = Ax_i(k) + Bu_i(k),\ i \in \mathbb{N} \\ x_0(k+1) = Ax_0(k) \end{cases}, \tag{1}$$

where $A \in \mathbb{R}^{n\times n}$ and $B \in \mathbb{R}^{n\times m}$ are the coefficient matrices, $u_i \in \mathbb{R}^m$ is the control input, $x_i \in \mathbb{R}^n$ is the states of the following agents, and $x_0 \in \mathbb{R}^n$ is the state of the leader agent. The goal is for all of the agents to have the same state as the leader:

$$\lim_{k\to\infty} ||x_i(k) - x_0(k)|| = 0,\ \forall i \in \mathbb{N}. \tag{2}$$

And the tracking error is defined as:

$$\eta = x(k) - \bar{x}_0(k), \tag{3}$$

where $x(k) = [x_1^T(k), x_2^T(k), x_3^T(k), \cdots, x_1^T(k)]^T$, and $\bar{x}_0(k) = 1_N \otimes x_0(k)$. However, each agent in distributed MASs can only know the information from surrounding neighbors, so the consensus error is defined as follows:

$$\varepsilon_i(k) = \sum_{J\in N_i} a_{ij}(x_j(k) - x_i(k)) + f_i(x_0(k) - x_i(k)),\ \forall i \in \mathbb{N}, \tag{4}$$

that is:

$$\varepsilon(k) = -\left((L+F)\otimes I_n\right)x(k) + \left((L+F)\otimes I_n\right)\bar{x}_0(k). \tag{5}$$

Lemma 2.1.[9] If matrix $(L+F)$ is non-singular, then the tracking error η is bounded:

$$||\eta(k)|| \leq (\delta_{\min}(L+F))||\varepsilon(k)||, \tag{6}$$

Remark 2.1. According to Lemma (2.1), the MASs synchronization can be achieved only by ensuring that the consensus error converges to 0.

If the model is known, the control policies can be designed as[4]:

$$u_i(k) = c(1+d_i+f_i)^{-1}K\varepsilon_i(k), \tag{7}$$

where c is the coupling gain, K is the feedback matrix, which can be designed according to model information. However, system modelling is a complex and time-consuming work in engineering, so data-driven model-free methods are urgently needed, and the proposed solution is one of this kind of method.

Assumption 2.1. The MAS's matrices (A, B) are linear time-invariant and unknown.

Assumption 2.2. The topology graph G contains at least a directed spanning tree, and at least a following agent can communicate with the leader.

Remark 2.2. If Assumption 2.2 is satisfied, the proposed method can achieve the goal of synchronization of the MAS (1). Besides, experiments show that the higher the degree of connectivity between the agents, the faster the synchronization will be achieved.

3. Synchronization Policies based on A3C

Since the state space and action space in MASs are high-dimensional data, BP neural networks are used to fit them to avoid the dimensionality disaster. Fig.1 shows the flowchart of the proposed solution, each part of it will be described separately below. First, a value function $Q(\varepsilon, a, \omega) = \phi(\varepsilon, a)^T\omega$ is designed to evaluate the value of error-action pair, and according to the idea of Temporal Difference Method[8], the Temporal Difference error (TD error) is defined as follows:

$$\delta = r_{k+1} + \gamma Q(\varepsilon_{k+1}, A_{k+1}) - Q(\varepsilon_k, A_k), \tag{8}$$

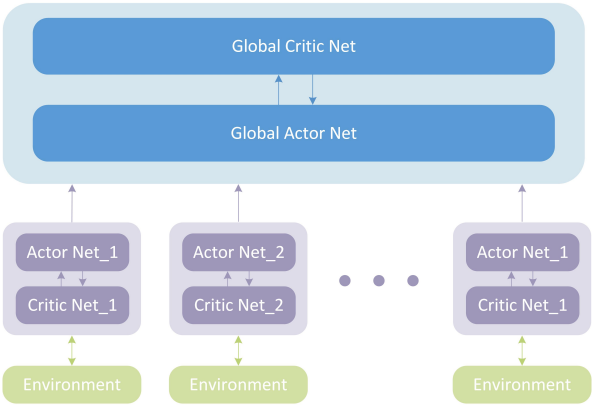

Fig. 1. The A3C based algorithm's flowchart in model-free leader-following MASs synchronization problem

where $r_{k+1} = \varepsilon_{k+1}^T \cdot \varepsilon_{k+1}$ is the reward at step $k+1$, γ is the discount factor. And the Mean Square Error (MSE) $L = \frac{1}{N}\delta^2$ is used as the update target for the value net $Q(\varepsilon, a, \omega)$. Thus it can be updated by the following equation:

$$\omega_{k+1} = \omega_k + \beta\delta\phi(\varepsilon_k, a_k), \tag{9}$$

where β is the learning rate.

The update aim of the actor net $\pi(\varepsilon, \theta) = \phi(\varepsilon)^T\theta$, also composed of the BP networks, is to generate policies that can maximize the value function $Q(\varepsilon, a, \omega)$:

$$J_{av}Q(\theta) = \sum_{\varepsilon} d^{\pi^\theta(\varepsilon)} \sum_{a} \pi_\theta(s)Q(\varepsilon, a, \omega), \tag{10}$$

where $d^{\pi^\theta(\varepsilon)}$ is the static distribution of the Markov chain generated by the policy π to the state s. Deriving this equation and approximating it with the Law of Large Numbers, we can get the following equation:

$$\nabla J(\theta) = E_{\pi(\theta)}\left[\nabla_\theta \log \pi_\theta(\varepsilon) Q_\pi(\varepsilon, a)\right]. \tag{11}$$

Besides,the Gaussian noise will be added to the output in order to sufficiently explore the action space:

$$\begin{aligned} &a \sim N(\mu(\varepsilon), \sigma^2) \\ &\nabla_\theta \log \pi_\theta(\varepsilon) = \tfrac{(a-\mu(\varepsilon))\phi(\varepsilon)}{\sigma^2}, \end{aligned} \tag{12}$$

where $\mu(\varepsilon) = \phi(\varepsilon)^T\theta$, the variance can be used as a tunable parameter to control the exploration rate. So the update equation is as follows:

$$\theta_{k+1} = \theta_k + \alpha\nabla_\theta \log \pi_\theta(\varepsilon_k)\left(r_{k+1} + \gamma Q(\varepsilon_{k+1}, A_{k+1}) - Q(\varepsilon_k, A_k)\right), \quad (13)$$

where α is the learning rate.

Since the data in the MASs environment are time-related, direct training may cause nets convergence difficult. DQN uses the Replay Buffer to store data and randomly extracts training to solve it, but excess storage space is not friendly to edge devices. However, the agents in MASs can interact with the environment at the same time, so they will learn and update the controller together. On the one hand, the distributed training approach breaks the data's time-related properties; On the other hand, the training process is substantially accelerated, making it more efficient.

4. Simulation Result

Considering a time-invariant leader-following system with a leader and 5 following agents, the system matrices are as follows:

$$A = \begin{bmatrix} -3 & 1 \\ -2 & 1 \end{bmatrix}, \quad B = \begin{bmatrix} 1 \\ 0 \end{bmatrix}. \quad (14)$$

The communication topology is shown in Fig.2. The communication matrix between the leader and following agents is $F = diag(1, 0, 0, 0, 0)$, and the Laplace matrix is as follows:

$$L = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & -1 & 0 & 0 \\ -1 & 0 & 2 & -1 & 0 \\ -1 & 0 & 0 & 2 & -1 \\ 0 & -1 & 0 & 0 & 1 \end{bmatrix}. \quad (15)$$

The learning curve is shown in Fig.3. As we can see, the return is initially very low, then gradually increases until it reaches 0 after around 20 iterations. The system's returns fluctuated over that due to distributed learning, and eventually stabilized. The tracking error and consensus error are shown in Fig.4 and Fig.5, respectively, the error swings a lot in the first 20 steps, but gradually decreases, indicating that the policy can synchronize MASs over time. Then the states of each agent are shown in Fig.6. After 20 steps, the following agents can oscillate synchronously with the leader, just like the tracking error and the consensus error. Finally, the energy

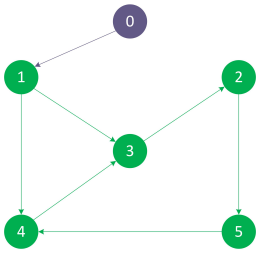

Fig. 2. The communication topology diagram of leader-following MASs

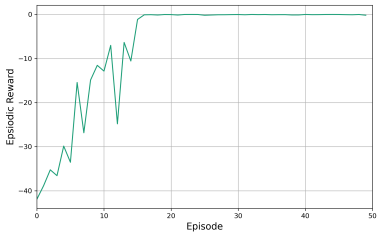

Fig. 3. The learning curve of leader-following MASs

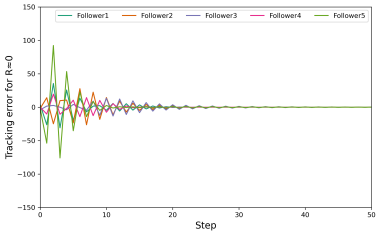

Fig. 4. The tracking error of leader-following MASs

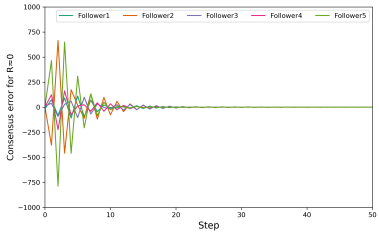

Fig. 5. The consensus error of leader-following MASs

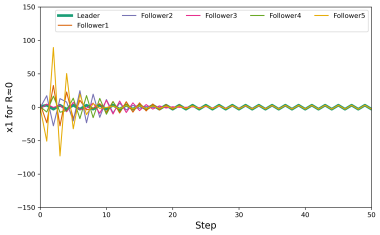

Fig. 6. The states of each agent in MASs

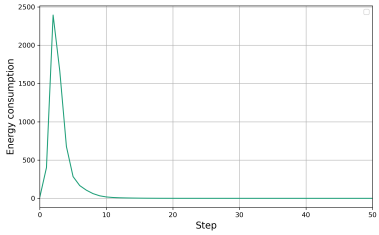

Fig. 7. The energy consumption of leader-following MASs

consumption is shown in Fig.7. In the first 10 steps, the controller needs to consume more energy to synchronize the system. After that, the energy usage is drastically reduced.

5. Conclusion

In this paper, a data-driven model-free synchronization solution is proposed for leader-following MASs. Concurrent training is used, which significantly

improves the training speed. Furthermore, while achieving synchronization, the proposed method can also minimize the energy consumption. Finally, a simple simulation is given to show the efficiency of this method. In the future, the model-free synchronization solution for continuous-time leader-following MASs will be considered based on reinforcement methods.

6. Acknowledgements

This work is supported by the Tianjin Natural Science Foundation of China (20JCYBJC01060) , the National Natural Science Foundation of China (62103203, 61973175), and the Fundamental Research Funds for the Central Universities, Nankai University(63221218) .

References

1. Y. Lai, R. Li, J. Shi and L. He, On the study of a multi-quadrotor formation control with triangular structure based on graph theory, Control Theory Appl 35, 1530 (2018).
2. G. Dudek, M. R. Jenkin, E. Milios and D. Wilkes, A taxonomy for multi-agent robotics, Autonomous Robots 3, 375 (1996).
3. M. Sader, Z. Liu, F. Wang and Z. Chen, Distributed robust fault-tolerant consensus tracking control for multi-agent systems with exogenous disturbances under switching topologies, International Journal of Robust and Nonlinear Control 32, 1618 (2022).
4. K. Hengster-Movric, K. You, F. L. Lewis and L. Xie, Synchronization of discrete-time multi-agent systems on graphs using riccati design, Automatica 49, 414 (2013).
5. Y. Zheng, Y. Zhu and L. Wang, Consensus of heterogeneous multi-agent systems, IET Control Theory & Applications 5, 1881 (2011).
6. K. Doya, Reinforcement learning in continuous time and space, Neural computation 12, 219 (2000).
7. H. Modares and F. L. Lewis, Optimal tracking control of nonlinear partially-unknown constrained-input systems using integral reinforcement learning, Automatica 50, 1780 (2014).
8. Y. Li, F. Wang, Z. Liu and Z. Chen, Leader-follower optimal consensus of discrete-time linear multi-agent systems based on q-learning, Proceedings of 2021 Chinese Intelligent Systems Conference , 492 (2022).
9. M. I. Abouheaf, F. L. Lewis, K. G. Vamvoudakis, S. Haesaert and R. Babuska, Multi-agent discrete-time graphical games and reinforcement learning solutions, Automatica 50, 3038 (2014).