JDCS
디지털콘텐츠학회논문지
Journal of Digital Contents Society
Vol. 23, No. 7, pp. 1277-1287, Jul. 2022

Check for updates

미지의 시변 지연을 갖는 다중 에이전트 시스템의 컨센서스를 위한 강화 학습

양 장 훈[1]
[1]서울미디어대학원대학교 인공지능 소프트웨어학과 부교수

Reinforcement Learning for the Consensus of a Multi-agent System with Unknown Time Varying Delays

Janghoon Yang[1]

[1]Associate Professor, Department of AI Software Engineering, Seoul Media Institute of Technology, Seoul 07590, Korea

[요 약]

자율 주행 이동체와 로봇 분야의 발전에 따라, 다중 에이전트 시스템의 컨센서스에 대한 관심이 증가하고 있다. 모델 기반의 컨센서스 알고리즘은 모델 정보가 제한적이고 불확실성이 존재하는 다양한 환경에 적용하는데 한계를 갖으며, 모델 기반의 강화학습 알고리즘은 수렴 속도가 느리거나, 파라미터 선택에 민감한 특성을 갖기 때문에, 이 알고리즘이 개발될 때에 가정한 일련의 환경에만 적용이 가능하다는 한계를 갖는다. 이러한 문제를 해결하기 위해서, 기존의 TD3[23] 강화 학습 알고리즘의 행동자 네트워크의 입력 레이어를 선형 함수를 취하고, 리워드 최적화를 적용하였다. 수치 모의 실험을 통해서 선학습된 가중치를 갖는 제안 강화학습 알고리즘이 모델 기반의 알고리즘과 유사한 성능을 갖으면서, 모델 기반 강화학습의 성능을 뛰어넘음을 확인하였다. 또한, 애블레이션 연구를 통하여, 기존의 강화 학습을 변형하는데 사용했던 2가지 방법이 안정적인 성능을 제공하는데 필수적임을 확인하였다.

[Abstract]

With the progress in autonomous vehicles and robots, there has been growing interest in the consensus of a multi-agent system. Limited model knowledge prevents the model-based consensus algorithm from being applicable to an environment with uncertainties while the model-based reinforcement learning (RL) algorithms are found to converge slowly and are applicable to a class of system environments they are assumed to be built on. To overcome these issues, TD3[23] was modified with a linear input layer at an actor network and reward shaping. Numerical simulation shows that the proposed RL with pretrained weight performs as well as the model-based one while it outperforms the model-based RL one. The ablation study also verifies the proposed modification is very critical in providing robust performance.

색인어 : 다중 에이전트 시스템, 컨센서스, 강화학습, 딥러닝, 지연

Keyword : Multi-agent system, Consensus, Reinforcement Learning, Deep Learning, Delay

http://dx.doi.org/10.9728/dcs.2022.23.7.1277

Received 27 May 2022; **Revised** 07 July 2022
Accepted 21 July 2022

***Corresponding Author; Janghoon Yang**

Tel: +82-2-6393-3237
E-mail: jhyang@smit.ac.kr

Ⅰ. Introduction

With the widespread use of the internet, more and more devices are connected over a network. Centralized control of multiple connected devices necessitates computational power and network bandwidth which are proportional to the number of devices. To deal with this issue, a distributed control has been studied. The distributed control reduces computational resources and network bandwidth by making a distributed decision from local data. An interesting distributed control is the consensus control of a multi-agent system (MAS), in which each agent tries to control itself to achieve a common goal from local information. With the growing interest in intelligent vehicles and machines, it has been applied to many different areas such as the formation of satellites [1], the coordination of multiple robots [2], freeway traffic congestion control [3], cooperative droop control for power sharing [4], and group decision making [5].

Local information transfer for a consensus control is bound to have a delay which depends on the communication protocol. The delay can incur instability of a MAS, slow down the convergence, or degrade control performance. Sufficient conditions for consensus-ability were shown to be a function of delay, packet drop, communication topology, and the dynamics of agents [6]. To deal with disagreement error incurred from transmission delay, an output feedback consensus control which exploits information on delay was developed for a homogeneous multiagent system with linear dynamics [7]. A distributed adaptive consensus control was proposed to provide a robust consensus control for MAS with unknown time delays and unknown bounded nonlinear dynamics [8]. A consensus control based on sliding mode control was also proposed to provide robust performance in the presence of unknown time varying delays and disturbances [9]. Various types of consensus control such as containment consensus control [10], an event-triggered control [11], and a group consensus control [12] were also studied for a MAS in the presence of delay. Some theoretical consensus algorithms were also experimentally validated in the presence of delay. Three two-wheeled robots were shown to achieve consensus when there was a delay of 0.5~1 seconds [13]. A heterogeneous MAS of two water level control test rigs with dissimilar dynamics and characteristics was shown to achieve the consensus with the aid of the articulated recursive prediction controller [14].

While most research on consensus control has focused on the model-based method, there has been growing interest in applying data-based methods such as reinforcement learning to overcome the limitation incurred by limited knowledge of the model of a system. An actor-critic neural network (NN) was exploited to implement the policy iteration algorithm which solves the coupled Hamilton-Jacobi-Bellman (HJB) equation for the optimal consensus control for a MAS with homogeneous linear dynamics [15]. An extra compensator was introduced to derive adaptive dynamic programming (ADP) based policy iteration algorithm for a MAS with nonlinear dynamics which approximated the value function with linear regression over the output of the neural network [16]. Adaptive dynamic programming (ADP) based RL algorithm which iteratively updates the Riccati matrix and feedback gain matrix was proposed for a MAS with linear dynamics. They developed the algorithms for both state feedback and output feedback [17]. The online RL algorithm for a MAS with heterogeneous linear dynamics was developed from formulating the consensus problem into a graphical minimax game and transforming the model-based policy iteration algorithm into an on-policy RL algorithm. [18]. An actor-critic neural network where the actor NN generates a control signal and the critic NN estimates the performance index function was exploited to implement a policy iteration based on optimal bipartite consensus control (OBCC) for a MAS with homogeneous linear dynamics [19]. An identification NN which estimates the next state from the current state was combined with an actor-critic network for a MAS with linear dynamics so that target action can be generated by using the output of the identification NN, which may help to stabilize the learning process [20]. An actor-critic NN with a simple single linear layer was developed to approximate the optimal consensus control for a second-order MAS [21]. It is observed that most existing research derives optimal consensus controls from the coupled HJB equation and approximates the performance index and action with NNs.

Recently, [22] proposed several deep learning based consensus algorithms. Some algorithms in [22] were shown to perform as well as the model-based one while they may have some performance degradation in some cases. [21] was also shown to converge very slowly in [22]. Most RL algorithms for consensus are model-based, which means that they may fail to perform well when they perform beyond the class the model they are assumed to be working on. Thus, this research adopts a state of art RL [23], which can be applicable to any domain without considering specific domain knowledge. Since optimizing parameters of the RL in [23] does not perform well, the RL was modified with two critical components, linear input layer at an actor network, and reward shaping. Simulation results verify that these two components play a critical role in providing reliable performance. The proposed algorithm is also found to perform as well as the model-based one for various system configurations.

This paper is organized as follows. A system model for a consensus of a second-order leader followers MAS will be given

with the goal of the consensus in section-2. In section-3, a brief summary will be provided for RL and twin-delayed deep deterministic policy. The modification of the [23] will be also explained. In section-4, the description of the simulation and simulation results which verify the efficiency of the proposed RL will be provided. Concluding remarks and future research will be given in section-5.

II. System Model and Problem Formulation

We consider a leader-followers MAS with a second-order dynamic. It consists of one leader and N followers. It is assumed that It operates in an environment with communication delays and disturbances. It is also assumed that these uncertainties are unknown. The corresponding system can be expressed as

$$\dot{v_i}(t) = u_i(t) + d_i(t) \tag{1}$$

$$x_i(t) = \int_{t_0}^{t} v(s)ds \tag{2}$$

where $v_i(t)$ is the velocity of the agent i, $u_i(t)$ is the control signal of the agent i, $d_i(t)$ is the disturbance at the agent i, $x_i(t)$ is the position of the agent i, and $\dot{v_i}(t)$ is the derivative of $v_i(t)$. The index for the leader agent is set to be 0. Several assumptions are made as follows to set the scope of this research. The perfect state information on its own state is available at each agent without measurement error. There is no packet loss in a communication channel. Communication topology does not change with time while it supports unidirectional communication.

The goal of consensus control for a leader-follower MAS can be given as

$$\lim_{t\to\infty}|x_i(t) - x_0(t)| = 0 \text{ for } i \in 1,2,...,N \tag{3}$$

That is, the difference between the position of the follower agent i and the leader agent 0 which is called disagreement error asymptotically goes to 0 through consensus control. When (3) is satisfied, the following condition for each agent i will hold too.

$$\lim_{t\to\infty}|x_i(t) - x_j(t)| = 0 \text{ for } j \in NBR_i \tag{4}$$

where NBR_i is the set of the indices of neighbor agents for the agent i. It is not possible to measure disagreement error at each agent, thus, local position error $e_i(t)$ which is often exploited for estimating performance or deriving control signal can be defined as follows.

$$e_i(t) = \sum_{j=0}^{N} a_{ij}(x_i(t) - x_j(t)) \tag{5}$$

where a_{ij} is the element of adjacency matrix which takes value 1 when the agent j is the neighbor of the agent i, otherwise 0.

III. Reinforcement Learning for Robust Consensus of A Multi-Agent System

3-1 Reinforcement Learning

RL is a learning algorithm which finds a policy to maximize the average return. When model information is available, an optimal policy can be solved through the Bellman equation. However, model information is often unavailable or partially available. RL tries to solve the Bellman equation without model information through trial and error. Let s be the state of a system. A policy $\pi(s)$ can be defined as a mapping from the state to action. The optimal RL policy $\pi^*(s)$ can be defined as follows.

$$\pi^*(s) = \arg max_{\pi(s)} E\{\mathrm{R}_0\} \tag{6}$$

where $R_t = \sum_{p=t}^{T} \gamma^p r(s_t, a_t)$ is the discounted sum of rewards called as return, s_t is a state at time t, a_t is an action generated by $\pi(a_t)$, $r(s_t, a_t)$ is a reward, and γ is a discounting factor determining how much future reward will be considered.

One of the most common network structures realizing RL is to take the form of an actor-critic network where an actor network predicts an action while a value network predicts the value of a state. The actor network A_θ usually updates its parameter with a policy gradient algorithm which takes a gradient of $E\{\mathrm{R}_0\}$ while the critic network V_ψ updates its parameter so that it can minimize the square of temporal difference error.

3-2 Twin Delayed Deep Deterministic Policy

Two main issues in the implementation of RL with an actor-critic network are overestimation of action state value function and variance of value estimate. These two problems often result in poor performance or instability in convergence with learning. To deal with these problems, the twin delayed deep deterministic policy algorithm (TD3) [23] was exploited. TD3 reduces over-estimation by taking the minimum between two action-value estimates, delaying the update of the target networks, and adding clipped noise to the target policy. TD3 was shown to

achieve the state of art performance over many OpenAI gym continuous control tasks, and Atari games. It is quite well known that most RL algorithms are sensitive to parameterization which requires tailoring to a specific problem. It was observed that simply applying TD3 to a consensus problem scarcely succeeded in deriving a good consensus algorithm with heuristic parameter adjustments.

3-3 Actor Network with a Linear Input Layer and Reward Shaping

To exploit TD3 as a baseline algorithm for the consensus of MAS, two modifications were introduced. The first modification is to use a linear network for an actor network. The underlying intuition is that most model-based consensus controls are given as a linear function of local errors. This implies that the preservation of phase information may have an important role in deriving consensus. The second modification is to shape the reward properly. There can be an infinite number of ways in defining rewards. The performance of the RL is dependent on reward design. One of the most naive ways to define rewards can be given as

$$r_{i,naive}(t) = -(|e_i(t)| + \alpha|\dot{e}_i(t)| + \beta|u_i(t)|) \tag{7}$$

where α is a weight for local velocity error and β is a weight for control signal. The local velocity error is considered since it is 0 with perfect consensus. That is each agent may move at the same velocity once they achieve the position consensus. The magnitude of the control signal is considered to prevent excessive control effort. When designing a reward, a huge penalty is assigned to an action resulting in a significant loss to an agent such as reaching the dead end or loss of a game. However, this linear loss does not assign a particularly large penalty to the large local error. Thus, with this reward, RL may not distinguish the case consisting of the majority of small local errors and a few large errors and one consisting of marginal errors. To deal with these issues. a reward is shaped from the naive reward as follows.

$$r_i(t) = 100(e^{r_{i,naive}(t)} + e^{0.1\times r_{i,naive}(t)} + e^{0.01\times r_{i,naive}(t)}) \tag{8}$$

Applying an exponential function to the naive reward assigns a large penalty to the large local error. However, when the naive rewards are less than a specific value, they are close to zero, which means that it may fail to distinguish a large error and a very large error. To deal with this issue, the second term and the third term are added so that the reward (8) has the power to distinguish a large error and a very large error while it assigns the large reward to an action resulting in a small error. The effect of the proposed designs will be assessed through an ablation study later in a subsequent section.

Ⅳ. Numerical Experiments

Numerical experiments consist of two parts. First, some key parameters will be determined from a heuristic search. Then, the proposed RL for consensus will be compared with three states of art methods, a model-based algorithm, a model-based RL, and supervised learning based algorithm.

4-1 The Configuration of the Proposed RL

After trying several different combinations of learning rates, the same learning rate of 0.0025 was set for both the actor and the critic networks. The update rate of both target networks was set 0.001. The update period for the actor network, the target actor network, and the target critic network was set 2 as used in [22]. The standard deviation of noise for the target policy smoothing was set as 2. This noise was clipped to (-3,3). The actor network has one hidden layer while the output activation function is set to be a hyper-tangent function. The critic network concatenates outputs of the separate subnetworks for state input, and action input, and adds two more hidden layers with Relu activation functions. The output layer of the critic network with a single node has no activation function. The size of the buffer for storing the experience was set as 500000. For exploration, noise following a normal distribution of mean 0 and standard deviation 2/(1+0.0005*episode) which decreases with the episodes, was added to the actor output. Adam optimizer was exploited for both networks.

To implement the proposed RL, α and β in (7) need to be determined. To this end, the performances of the proposed RL for the different values of α and β were evaluated for a MAS with simulations. The number of agents was 5. The communication graph A in figure-1 was adopted. In this graph, agent 1 receives information from two neighbors while other follower agents receive it from one neighbor. The total simulation time is 41 seconds with a sampling period of 0.01 seconds. Disturbance, communication delays, and the acceleration of the leader agent, were set as $d_i(t) = \sin(11t) + \cos(13t)$, $\tau_{ij}(t) = 0.25(\cos(t + (i+j)\pi/7) + 1)$, and $\dot{v_0}(t) = \cos(3t) + \cos(7t)$. For the first 10 seconds, the acceleration of the follower agents was set to be disturbance only so that delayed information exchange can be implemented during the remaining periods. State information was defined as the most recent 5 local position errors and local velocity errors which make the dimension of the state 10. The initial position of each agent

was generated from the standard normal distribution. For the remaining 31 seconds, the proposed RL was implemented. The 31 seconds was chosen since model-based consensus was observed to converge in most cases within 30 seconds and 1 second was used for performance evaluation.

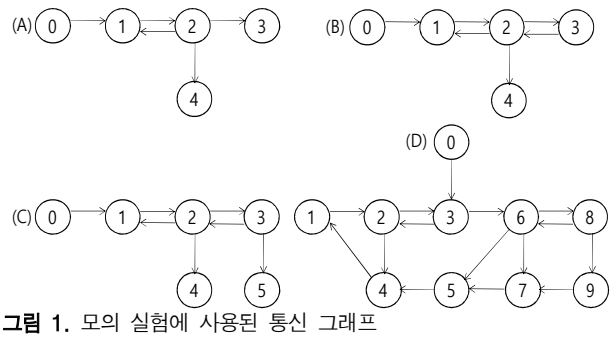

그림 1. 모의 실험에 사용된 통신 그래프
Fig. 1. Communication graphs used for simulation

To determine the parameters, α and β in (7), α was first determined with setting $\beta=0$. For the last 50 episodes among total of 200 episodes, the sample averages of the square of local error and disagreement error were calculated during the last 1 second for each episode. The mean square of local error (MSE), and the mean square of disagreement (MSD) were calculated from averaging out these 50 values. Table-1 shows that α of 0.5 provides the best MSE while α of 0.1 provides the best MSD. It is observed that when α is too small or too large, it incurs instability in convergence. α of 0.5 was finally chosen. After fixing α as 0.5, β was determined from evaluating the performance over different values as found in table-2. For small β, the performances are found to be similar. However, when β is large, it often diverges. Since the proposed algorithm provides the robust performance for small β, β was chosen as 0.

4-2 Performance Evaluation of the Proposed RL

After determining α and β in (7), the performances of the proposed RL were evaluated for several different system configurations which are summarized in table-3 and table-4. The accelerations of the leader agent are either periodic or have slowing increasing envelop with some upper bound. Disturbances are designed such that each agent has the same type of bounded disturbance with different magnitudes at the same instance. Communication graphs with varying degrees of nodes with a maximum degree of 2 are considered for the purpose of the simplicity of simulation. Bounded delays are designed such that they can be different for different communication links and they can have different speeds of variation over time. It is observed that even though the considered system configurations do not cover all possible system configurations, they are articulated such that the performance characteristics of the proposed algorithm can be assessed for various system configurations. Other simulation setups will be the same as one in section 4-1, unless otherwise stated.

표 1. α 값에 따른 MSE와 MSD (x: 발산 발생)
Table 1. MSE and MSD for different αs (x : occurrence of divergence)

α	MSE	MSD
0.001	x	x
0.005	x	x
0.01	11.49	40.90
0.05	15.58	42.12
0.1	0.61	21.94
0.5	0.01	22.88
1	13.38	29.77
5	x	x

표 2. β 값에 따른 MSE와 MSD (x: 발산 발생)
Table 2. MSE and MSD for different βs (x : occurrence of divergence)

β	MSE	MSD
0	0.01	22.88
0.01	0.02	22.05
0.05	0.04	18.37
0.1	0.02	19.63
0.5	0.03	19.76
1	x	x
5	x	x
10	x	x

표 3. 성능 테스트를 위한 리더 에이전트의 가속도와 왜란
Table 3. Configuration of the acceleration of the leader agent and the disturbance for testing

case	Acceleration of leader agent	Disturbance
1	$\cos(7t)+\cos(3t)$	$\sin(11t+k)+\cos(13t+k)$
2	$[\cos(7t)+\cos(3t)](2-e^{-t})$	$\sin(11t+k)(3-e^{-t})$
3	$[\cos(7t)+\cos(3t)](2-e^{-t})$	$\sin(11t+k)(3-e^{-t})$
4	$[\cos(7t)+\cos(3t)](2-e^{-t})$	$\sin(11t+k)(3-e^{-t})$
5	$[\cos(7t)+\cos(3t)](2-e^{-t})$	$\sin(11t+k)(3-e^{-t})$
6	$\cos(17t)(3-e^{-t})(2+\cos(13t))^{-1}$	$\cos(23t+k)(e^{-0.1t}+1)-e^{-t}$

표 4. 성능 테스트를 위한 통신 그래프와 지연
Table 4. Configuration of communication graphs and delays for testing

case	Graph	Delay
1	A	$0.25(1+\cos(t+(k+l)\pi/7))$
2	A	$0.25(1+\cos(111t+(k+l)\pi/7))$
3	B	$0.5(1+e^{-0.1(k+l)t})^{-1}$
4	C	$0.5(1-(1+e^{-0.1(k+l)t})^{-1})$
5	D	$0.5(1+\cos(t+(k+l)\pi/7))(2+e^{=0.1(k+l)t})^{-1}$
6	D	$0.5(1-0.5(1+\cos(t+(k+l)\pi/7))e^{-0.1(k+l)t})$

We compare the proposed RL with three existing states of art algorithms, a model-based consensus algorithm with sliding mode control [9], deep-learning based algorithm with supervised learning [22], and the model-based RL [21]. Several different algorithms for deep-learning based algorithms with supervised learning were introduced in [22]. Among them, disjoint DL and joint DL were chosen for comparison since they showed the best performance. These algorithms are different in network structures generating a control signal. Please refer to [22] for details. Two different versions of RL types algorithms are also considered. They are different in initialization. The parameter of the first one was initialized randomly while the second one was initialized with the pretrained weight. To distinguish them, "(p)" was added to the end of the name of the second type algorithm. The pretrained weight was obtained by training the RL algorithm for the system configuration described in section 4-1.

Figures 2-7 show the MSE of the local error for system configurations in table-3 and table-4. These plots were obtained by 20 realizations for each case. The model-based algorithm is observed to achieve the best average MSE for all cases. The proposed RL with random initialization is found to struggle to learn. However, the proposed RL with pretrained weight achieves the second best average MSE while it does not have model knowledge. Even though the proposed RL(p) has a similar average MSE to that of the model-based RL(p) for several cases, the proposed RL(p) is found to provide consistent performance for different position initializations, which is evidenced by very small variance for all cases. In addition, the proposed RL(p) is shown to provide robust performance for different system configurations. It achieves the MSE of 0.01 or so for all cases while the average MSEs of other algorithms have a dependency on system configuration.

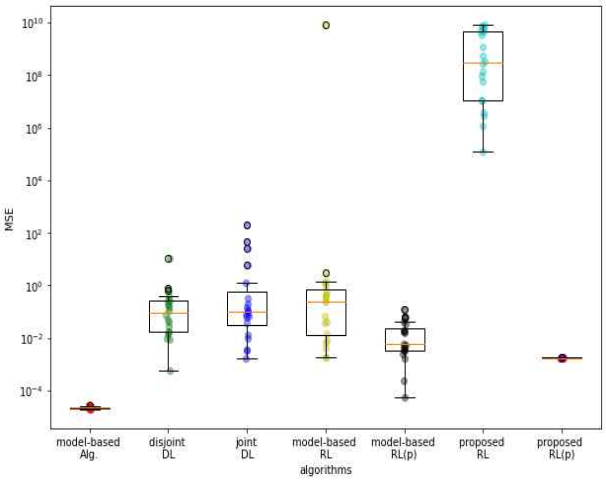

그림 2. 시스템 설정 1에 따른 지역 오류의 MSE
Fig. 2. MSE of local error for system configuration 1

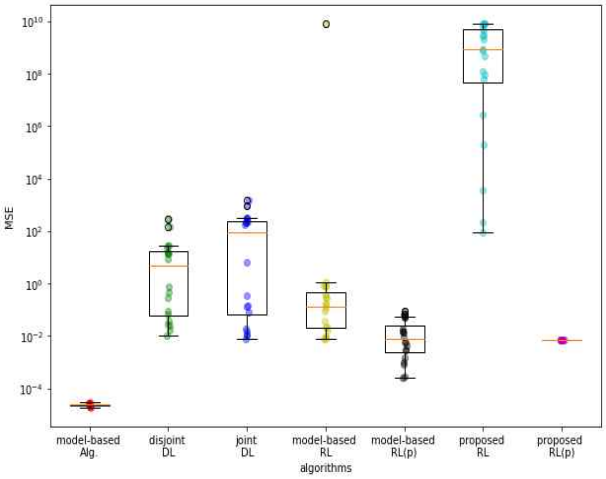

그림 3. 시스템 설정 2에 따른 지역 오류의 MSE
Fig. 3. MSE of local error for system configuration 2

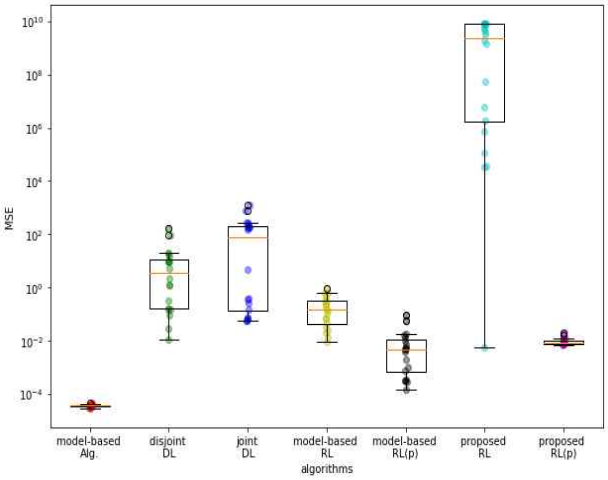

그림 4. 시스템 설정 3에 따른 지역 오류의 MSE
Fig. 4. MSE of local error for system configuration 3

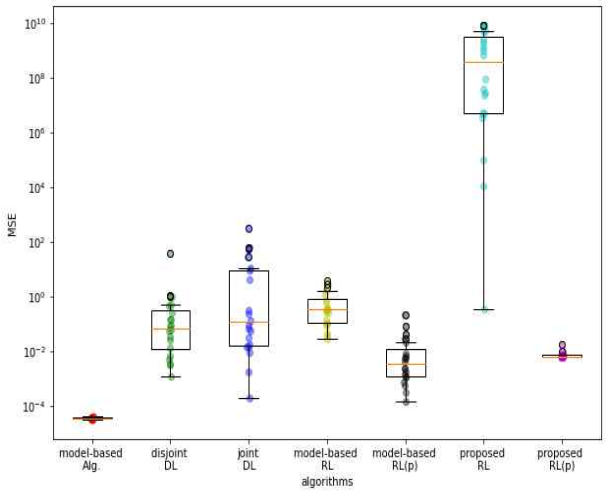

그림 5. 시스템 설정 4에 따른 지역 오류의 MSE
Fig. 5. MSE of local error for system configuration 4

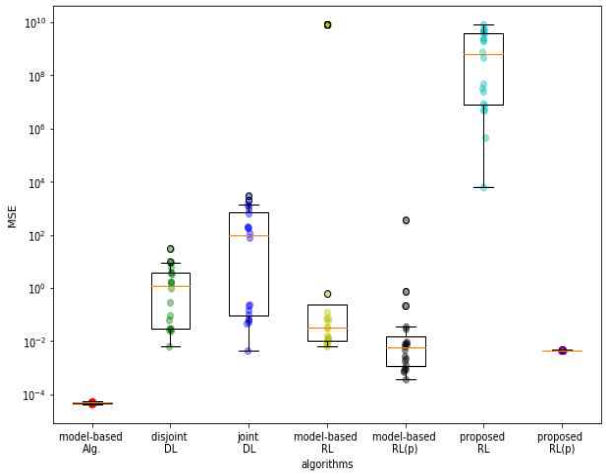

그림 6. 시스템 설정 5에 따른 지역 오류의 MSE
Fig. 6. MSE of local error for system configuration 5

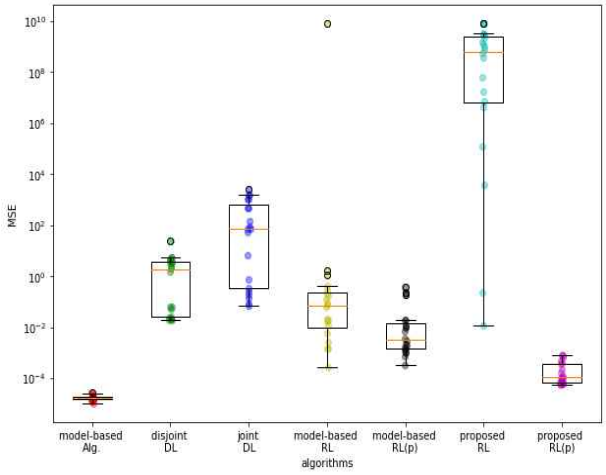

그림 7. 시스템 설정 6에 따른 지역 오류의 MSE
Fig. 7. MSE of local error for system configuration 6

Figures 8-13 show the MSDs of the disagreement error for system configurations in table-3 and table-4. The characteristics of the MSEs of the disagreement error are pretty much similar to those of the MSEs for the local error in most cases. However, it is noted that the average MSD is relatively larger than the average MSE for all algorithms and all cases. This is quite natural in the sense that the algorithm is optimized to reduce MSE and there is no direct access to MSD due to delay. To evaluate the performance quantitatively, the ratios of MSE and MSD less than a specified value were adopted, since the average value does not provide meaningful implications due to the values which were distributed very widely.

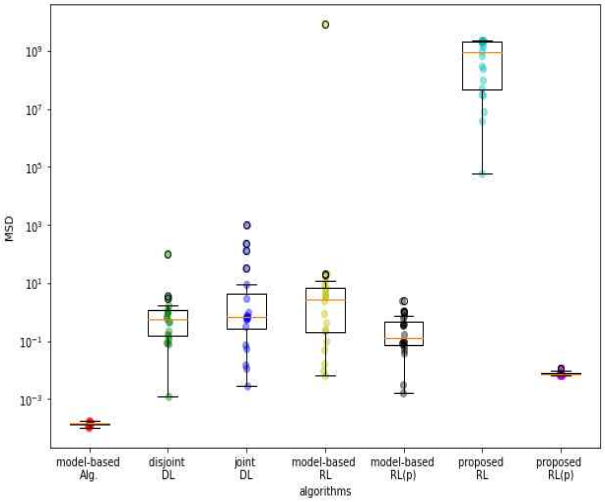

그림 8. 시스템 설정 1에 따른 불일치 오류의 MSD
Fig. 8. MSD of disagreement error for system configuration 1

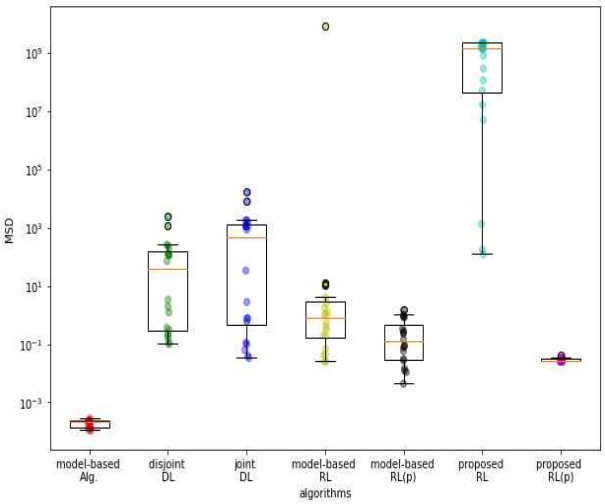

그림 9. 시스템 설정 2에 따른 불일치 오류의 MSD
Fig. 9. MSD of disagreement error for system configuration 2

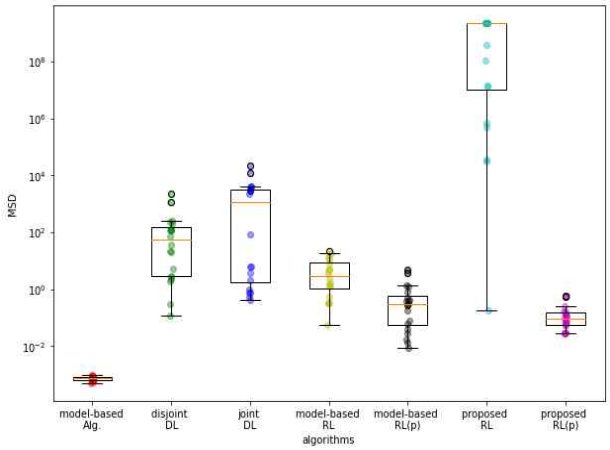

그림 10. 시스템 설정 3에 따른 불일치 오류의 MSD
Fig. 10. MSD of disagreement error for system configuration 3

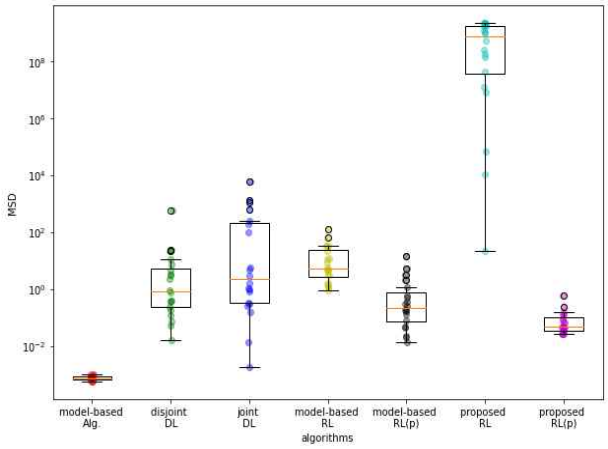

그림 11. 시스템 설정 4에 따른 불일치 오류의 MSD
Fig. 11. MSD of disagreement error for system configuration 4

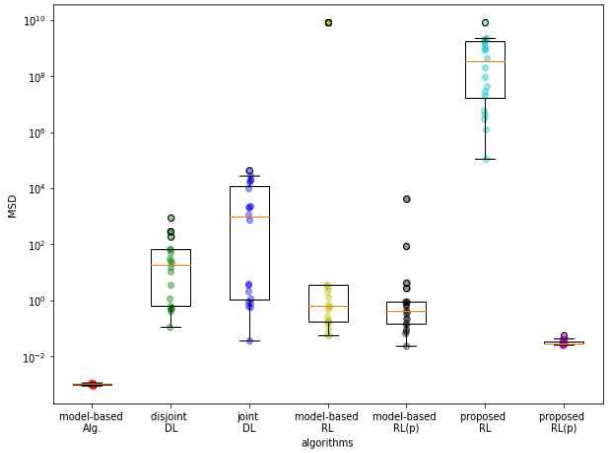

그림 12. 시스템 설정 5에 따른 불일치 오류의 MSD
Fig. 12. MSD of disagreement error for system configuration 5

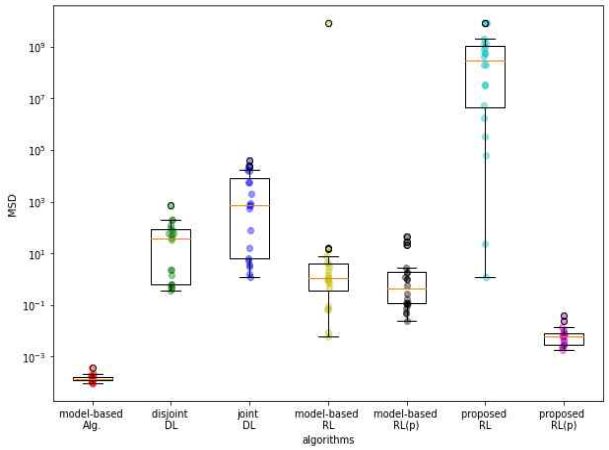

그림 13. 시스템 설정 6에 따른 불일치 오류의 MSD
Fig. 13. MSD of disagreement error for system configuration 6

The ratios of MSE and MSD less than 0.1 are presented in table-5. The proposed RL(p) and the model-based algorithm provide the ratio of 1 for MSE less than 0.1 for all cases while the ratios achieved by the proposed RL(p) are less than those by the model-based one for other 2 cases. Nonetheless, the proposed RL(p) provides a ratio larger than or equal to those of other algorithms than the model-based one in MSE and MSD. Table-6 which presents the ratios of MSE and MSD less than 1 clearly shows that the proposed algorithm provides the same performance as the model-based one while outperforming other algorithms. As can be seen in the preceding figures, the proposed RL(p) provides a robust performance to the initialization and system environment in comparison to the model-based RL(p).

To assess the effect of the components newly introduced to the proposed RL, an ablation study was performed. For the system configuration described in section 4-1, 50 realizations were performed with 100 episodes for each realization. Figure-14 shows the convergence characteristics with increasing episodes. This figure clearly shows that the proposed RL increases the average reward from exploiting the linear input layer at the action network. In addition, the linear input layer is observed to accelerate the convergence. The reward shaping is found to contribute to increasing the convergence speed. Even though the average reward increases very slowly without reward shaping, even the average reward value is observed to fluctuate with episodes, which implies that a convergence problem may happen. Table-7 shows the quantitative evaluation of the introduced two components in terms of the ratios of MSE less than the threshold. Input linear layer is found to be a more critical component to the performance while the contribution of the reward shaping is significant. These results verify that the proposed RL may not perform well without either of them.

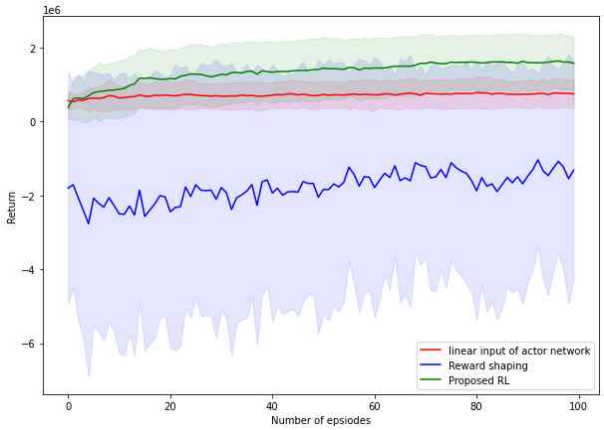

그림 14. 새롭게 추가된 요소에 의한 수렴 특징
Fig. 14. The convergence characteristics with ablation study

표 5. 0.1보다 작은 MSE와 MSD의 비율 (괄호 안의 값이 0.1보다 작은 MSD의 비율)
Table 5. The ratios of MSE and MSD less than 0.1 (The value in parenthesis is the ratio of MSD less than 0.1)

case	1	2	3	4	5	6
model-based alg.	1.00 (1.00)	1.00 (1.00)	1.00 (1.00)	1.00 (1.00)	1.00 (1.00)	1.00 (1.00)
disjoint DL	0.50 (0.20)	0.35 (0.00)	0.15 (0.00)	0.60 (0.15)	0.40 (0.00)	0.45 (0.00)
joint DL	0.50 (0.25)	0.3 (0.15)	0.25 (0.00)	0.50 (0.10)	0.30 (0.05)	0.10 (0.00)
model-based RL	0.45 (0.25)	0.45 (0.25)	0.40 (0.05)	0.25 (0.00)	0.70 (0.10)	0.60 (0.25)
model-basedRL (p)	0.95 (0.50)	1.00 (0.50)	1.00 (0.35)	0.95 (0.30)	0.85 (0.20)	0.85 (0.20)
proposed RL	0.00 (0.00)	0.00 (0.00)	0.05 (0.00)	0.00 (0.00)	0.00 (0.00)	0.05 (0.00)
proposed RL(p)	1.00 (1.00)	1.00 (1.00)	1.00 (0.55)	1.00 (0.75)	1.00 (1.00)	1.00 (1.00)

표 6. 1보다 작은 MSE와 MSD의 비율 (괄호 안의 값이 1보다 작은 MSD의 비율)
Table 6. The ratio of MSE and MSD less than 1 (The value in parenthesis is the ratio of MSD less than 1)

case	1	2	3	4	5	6
model-based alg.	1.00 (1.00)	1.00 (1.00)	1.00 (1.00)	1.00 (1.00)	1.00 (1.00)	1.00 (1.00)
disjoint DL	0.95 (0.70)	0.50 (0.35)	0.35 (0.10)	0.85 (0.55)	0.50 (0.30)	0.45 (0.30)
joint DL	0.75 (0.70)	0.45 (0.40)	0.45 (0.25)	0.65 (0.40)	0.45 (0.25)	0.35 (0.00)
model-based RL	0.80 (0.45)	0.90 (0.50)	1.00 (0.25)	0.75 (0.05)	0.80 (0.55)	0.85 (0.45)
model-basedRL (p)	1.00 (0.85)	1.00 (0.90)	1.00 (0.80)	1.00 (0.75)	0.95 (0.80)	1.00 (0.60)
proposed RL	0.00 (0.00)	0.00 (0.00)	0.05 (0.05)	0.05 (0.00)	0.00 (0.00)	0.10 (0.00)
proposed RL(p)	1.00 (1.00)	1.00 (1.00)	1.00 (1.00)	1.00 (1.00)	1.00 (1.00)	1.00 (1.00)

표 7. 새롭게 추가된 요소에 의한 임계치 보다 작은 MSE의 비율
Table 7. The ratio of MSE less than threshold with ablation study

Threshold	0.1	1
linear activation	0.06	0.06
reward shaping	0.1	0.26
proposed RL.	0.64	0.7

V. Conclusions

The existing RL algorithms often experience sensitivity to the parameterization and performance degradation from the slow convergence when they are applied to the consensus of a MAS. In this paper, an RL for the consensus of the second-order leader-followers MAS was proposed to overcome these issues. The proposed RL benefits from the existing articulated structures of TD3, the liner input layer at an actor network, reward shaping, and pretraining. The proposed RL with pretrained weight was shown to provide consistent performance with different initializations for various system configurations. It also achieves a performance comparable to the model-based one, while it outperformed the existing model-based RL. These results imply that the proposed RL with pretrained weight converged for all simulation cases thanks to the pretrained weight.

There are a few further works to be addressed in future research. Even though the proposed RL with pretrained weight performed well, the proposed RL with random initialization had difficulty in learning. On the contrary, the model-based RL with random initialization had a few successes. This implies that there can be a chance to improve the proposed RL through exploiting some structures of the model-based RL. To implement the proposed RL in a practical system, the model needs to be scaled down. Thus the distilled model with a controlled loss will be of interest.

Acknowledgements

This work was supported by the Ministry of Education of the Republic of Korea and the National Research Foundation of Korea (NRF-2020S1A5A2A03045921)

이 논문은 2020년 대한민국 교육부와 한국연구재단의 지원을 받아 수행된 연구임 (NRF-2020S1A5A2A03045921)

References

[1] Z. Li, Z. Duan, G. Chen and L. Huang, “Consensus of Multiagent Systems and Synchronization of Complex Networks: A Unified Viewpoint,” *IEEE Transactions on Circuits and Systems I: Regular Papers,* Vol. 57, No. 1, pp. 213-224, Jan. 2010. doi: 10.1109/TCSI.2009.2023937.

[2] V. Trianni, D. De Simone, A. Reina and A. Baronchelli, “Emergence of Consensus in a Multi-Robot Network: From Abstract Models to Empirical Validation,” *IEEE Robotics and Automation Letters*, Vol. 1, No. 1, pp. 348-353, Jan. 2016. doi: 10.1109/LRA.2016.2519537.

[3] B. Kim and H. Ahn, “Distributed Coordination and Control for a Freeway Traffic Network Using Consensus Algorithms,” *IEEE Systems Journal*, vol. 10, no. 1, pp. 162-168, March 2016. doi: 10.1109/JSYST.2014.2318054.

[4] J. Zhou, M. Tsai and P. Cheng, “Consensus-Based Cooperative Droop Control for Accurate Reactive Power Sharing in Islanded AC Microgrid,” *IEEE Journal of Emerging and Selected Topics in Power Electronics,* Vol. 8, No. 2, pp. 1108-1116, June 2020. doi: 10.1109/JESTPE.2019.2946658.

[5] Z. Zhang, Z. Li, and Y. Gao, “Consensus reaching for group decision making with multi-granular unbalanced linguistic information: A bounded confidence and minimum adjustment-based approach,” *Information Fusion*, Vol. 74, pp. 96-110, Oct. 2021. https://doi.org/10.1016/j.inffus.2021.04.006.

[6] J. Zheng, L. Xu, L. Xie and K. You, "Consensusability of Discrete-Time Multiagent Systems With Communication Delay and Packet Dropouts," *IEEE Transactions on Automatic Control*, Vol. 64, No. 3, pp. 1185-1192, March 2019. doi: 10.1109/TAC.2018.2846679.

[7] C. Wang, Z. Zuo, Z. Qi and Z. Ding, "Predictor-Based Extended-State-Observer Design for Consensus of MASs With Delays and Disturbances," *IEEE Transactions on Cybernetics*, Vol. 49, No. 4, pp. 1259-1269, April 2019. doi: 10.1109/TCYB.2018.2799798.

[8] B. Cui, T. Ma, F. L. Lewis, C. Zhao, Y. Song, and C. Feng, “Distributed adaptive consensus control of heterogeneous multi-agent chaotic systems with unknown time delays,” *IET Control Theory & Applications,* Vol.9, No. 16, pp. 2414-2422, Oct. 2015. DOI: 10.1049/iet-cta.2015.0187.

[9] J. Yang, "A Consensus Control for a Multi-Agent System With Unknown Time-Varying Communication Delays," *IEEE Access*, Vol. 9, pp. 55844-55852, 2021. doi: 10.1109/ACCESS.2021.3070388.

[10] D. Wang, N. Zhang, J. Wang and W. Wang, "Cooperative Containment Control of Multiagent Systems Based on Follower Observers With Time Delay," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, Vol. 47, No. 1, pp. 13-23, Jan. 2017. doi: 10.1109/TSMC.2016.2577578.

[11] G. Duan, F. Xiao and L. Wang, "Asynchronous Periodic Edge-Event Triggered Control for Double-Integrator Networks With Communication Time Delays," *IEEE Transactions on Cybernetics*, Vol. 48, No. 2, pp. 675-688, Feb. 2018. doi: 10.1109/TCYB.2017.2651026.

[12] G. Wen, Y. Yu, Z. Peng, and H. Wang, "Dynamical group consensus of heterogenous multi-agent systems with input time delays," *Neurocomputing*, Vol. 175, Part A, pp. 278-286, Jan. 2016. https://doi.org/10.1016/j.neucom.2015.10.060

[13] W. Qiao and R. Sipahi, "Consensus Control Under Communication Delay in a Three-Robot System: Design and Experiments," *IEEE Transactions on Control Systems Technology*, Vol. 24, No. 2, pp. 687-694, March 2016. doi: 10.1109/TCST.2015.2458776.

[14] N. A. M. Subha and G. -P. Liu, "Design and Practical Implementation of External Consensus Protocol for Networked Multiagent Systems With Communication Delays," *IEEE Transactions on Control Systems Technology*, Vol. 23, No. 2, pp. 619-631, March 2015. doi: 10.1109/TCST.2014.2341617.

[15] H. Zhang, H. Jiang, Y. Luo and G. Xiao, "Data-Driven Optimal Consensus Control for Discrete-Time Multi-Agent Systems With Unknown Dynamics Using Reinforcement Learning Method," *IEEE Transactions on Industrial Electronics*, Vol. 64, No. 5, pp. 4091-4100, May 2017. doi: 10.1109/TIE.2016.2542134.

[16] J. Zhang, H. Zhang and T. Feng, "Distributed Optimal Consensus Control for Nonlinear Multiagent System With Unknown Dynamic," *IEEE Transactions on Neural Networks and Learning Systems*, Vol. 29, No. 8, pp. 3339-3348, Aug. 2018. doi: 10.1109/TNNLS.2017.2728622.

[17] X. Wang, and H. Su, “Completely model-free RL-based consensus of continuous-time multi-agent systems,” *Applied Mathematics and Computation*, Vol.382, pp.1-11, Oct. 2020. https://doi.org/10.1016/j.amc.2020.125312.

[18] W. Dong, C. Wang, J. Li and J. Wang, "Graphical Minimax Game and On-Policy Reinforcement Learning for Consensus of Leaderless Multi-Agent Systems," in *Proceeding of the 16th International Conference on Control & Automation (ICCA)*, Japan, pp. 606-611, 2020. doi: 10.1109/ICCA51439.2020.9264527.

[19] Z. Peng, J. Hu, K. Shi, R. Luo, R. Huang, B. K. Ghosh, and J. Huang, "A novel optimal bipartite consensus control scheme for unknown multi-agent systems via model-free reinforcement learning," *Applied Mathematics and Computation*, Vol.369, pp.1-14, Mar. 2020. https://doi.org/10.1016/j.amc.2019.124821.

[20] Y. Liu, T. Li, Q. Shan, R. Yu, Y. Wu, and C.L.P. Chen, "Online optimal consensus control of unknown linear multi-agent systems via time-based adaptive dynamic programming," *Neurocomputing*, Vol.404, pp.137-144, Sept. 2020. https://doi.org/10.1016/j.neucom.2020.04.119.

[21] J. Li, L. Ji, and H. Li, "Optimal consensus control for unknown second-order multi-agent systems: Using model-free reinforcement learning method," *Applied Mathematics and Computation*, Vol.410, pp.1-15, Dec. 2021. https://doi.org/10.1016/j.amc.2021.126451.

[22] J. Yang, "Deep Learning-Based Consensus Control of a Multi-Agents System with Unknown Time-Varying Delay," *Electronics*, Vol. 11, No.8, pp.1-15, Apr. 2022. https://doi.org/10.3390/electronics11081176.

[23] S. Fujimoto, H. van Hoof, and D. Meger, "Addressing Function Approximation Error in Actor-Critic Methods," in *Proceeding of the* 35*th International Conference on Machine Learning,* Stockholm, Sweden, pp. 1282-1291, 2018.

양장훈(Janghoon Yang)

2001년 : University of Southern California (공학박사)

2001년～2006년: 삼성전자, 책임연구원
2006년～2010년: 연세대학교, 연구교수
2010년～현 재: 서울미디어대학원 뉴미디어학부, 부교수
※관심분야 : 인공지능, 중재 기술, 감성 공학, 간사이 공학, 정보이론, 이종 시스템 제어, 무선통신, 무선 네트워크