JANT 한국항행학회논문지
Journal of Advanced Navigation Technology

응용 및 융합 기술

J. Adv. Navig. Technol. 26(4): 226-234, Aug. 2022

다중 에이전트 시스템의 컨센서스를 위한 슬라이딩 기법 강화학습

A slide reinforcement learning for the consensus of a multi-agents system

양 장 훈
서울미디어대학원대학교 인공지능 응용소프트웨어학과

Janghoon Yang
Department of AI Software Engineering, Seoul Media Institute of Technology, Seoul, 07590, Korea

[요 약]

자율 주행체와 네트워크 기반 제어 기술의 발달에 따라서, 하나의 에이전트를 제어하는 것을 넘어서 다수의 이동체를 분산 제어하는데 사용 가능한 다중 에이전트의 컨센서스 제어에 대한 관심과 연구가 증가하고 있다. 컨센서스 제어는 분산형 제어이기 때문에, 정보 교환은 실제 시스템에서 지연을 가지게 된다. 또한, 시스템에 대한 모델을 정확히 수식적으로 표현하는데 있어서 한계를 갖는다. 이런 한계를 극복하는 방법 중에 하나로서 강화 학습 기반 컨센서스 알고리즘이 개발되었지만, 불확실성이 큰 환경에서 느린 수렴을 갖는 경우가 자주 발생하는 특징을 보이고 있다. 따라서, 이 논문에서는 불확실성에 강인한 특성을 갖는 슬라이딩 모드 제어를 강화학습과 결합한 슬라이딩 강화학습 알고리즘을 제안한다. 제안 알고리즘은 기존의 강화학습 기반 컨센서스 알고리즘의 제어 신호에 슬라이딩 모드 제어 구조를 추가하고, 시스템의 상태 정보를 슬라이딩 변수를 추가하여 확장한다. 모의실험 결과 다양한 시변 지연과 왜란에 대한 정보가 주어지지 않았을 때에 슬라이딩 강화학습 알고리즘은 모델 기반의 알고리즘과 유사한 성능을 보이면서, 기존의 강화학습에 비해서 안정적이면서 우수한 성능을 보여준다.

[Abstract]

With advances in autonomous vehicles and networked control, there is a growing interest in the consensus control of a multi-agents system to control multi-agents with distributed control beyond the control of a single agent. Since consensus control is a distributed control, it is bound to have delay in a practical system. In addition, it is often difficult to have a very accurate mathematical model for a system. Even though a reinforcement learning (RL) method was developed to deal with these issues, it often experiences slow convergence in the presence of large uncertainties. Thus, we propose a slide RL which combines the sliding mode control with RL to be robust to the uncertainties. The structure of a sliding mode control is introduced to the action in RL while an auxiliary sliding variable is included in the state information. Numerical simulation results show that the slide RL provides comparable performance to the model-based consensus control in the presence of unknown time-varying delay and disturbance while outperforming existing state-of-the-art RL-based consensus algorithms.

Key word : Consensus, Delay, Multi-agents system, Reinforcement Learning, Sliding mode control.

https://doi.org/10.12673/jant.2022.26.4.226

Received 27 June 2022; **Revised** 1 August 2022
Accepted (Publication) 24 August 2022 (30 August 2022)

***Corresponding Author; Janghoon Yang**

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

Ⅰ. Introduction

With the rapid progress in network, computing, and control technologies, many devices are controlled over a network. They can be either controlled through a centralized processor, or each own decentralized processor. However, the centralized processor often experiences the limitation of the computational complexity with the growing number of agents in a system while the decentralized one often experiences the performance degradation due to the limited information. The distributed control has the tradeoff between the complexity and performance through information exchange with a subset of the objects in the system.

Depending on the operating environment, the goal of a distributed control can be different. Among many different control objectives, a consensus has been paid significant attention recently. The consensus control of multiple agents can be defined as controling each object to achieve a common goal. The consensus control has been applied to many different fields such as the coordination of robots [1], voltage and phase synchronization of power networks [2], the altitude formation of satellites [3], traffic control [4], group decision-making [5], and network server load balancing [6].

In consensus control, each agent exchanges information with neighbor agents. Thus, the structure of the graph representing communication among agents has a critical impact on control performance. The algebraic connectivity which is the second smallest eigenvalue of the Laplacian matrix was shown to determine the convergence speed of the consensus in a continuous time domain [7]. A consensus algorithm converging to the average of the initial state in a discrete-time was proposed and the algebraic connectivity was shown to determine the convergence of the corresponding consensus algorithm [8]. The Laplacian matrix was proven to be an optimal linear consensus algorithm for a multi-agent system (MAS) with the first-order dynamics in the perspective of a linear-quadratic-regulator (LQR) [9]. A hierarchical feedback controller based on LQR for the consensus of a leader-follower MAS was proposed to have a tradeoff between the complexity and performance through an articulated graph structure [10]. Communication to exchange information is bound to have delays. However, the delay in a control system can have an effect on the stabilization of the system if it is not considered properly in the development of the control algorithm [11]. A linear matrix inequality (LMI) was exploited to derive a sufficient condition for the average consensus of a MAS with higher-order dynamics in the presence of time-varying delays [12]. Using the Nyquist criterion, a consensus protocol was developed for a heterogenous MAS with bounded input delay [13]. A sliding mode control was exploited to develop a robust consensus control for a MAS with the second-order dynamics in the presence of unknown time delay and bounded disturbance [14]. A robust output feedback consensus control was developed from a sufficient condition for the secure consensus of a MAS to deal with sampling error and deny-of-service (DOS) attacks [15].

While most researches have developed an algorithm from an abstract mathematical model with assumptions and preconditions, perfect system information is often unavailable. To deal with this issue, a novel approach has been made with reinforcement learning (RL). Combined with deep learning, deep reinforcement learning has shown the potential to learn efficiently in many different areas such as computer games [16], controls [17], and robotics [18]. RL often tries to generate an action to maximize the return which is the expected cumulated reward. A popular method of realizing RL with a neural network (NN) is to introduce two networks, an actor-network to generate an action and a critic network to estimate the action-state value. A policy iteration algorithm for the consensus of a MAS with linear dynamics was developed from the coupled Hamilton-Jacobi-Bellman equation with an actor-critic NN [19]. Similarly, RL consensus algorithms were designed by solving algebraic Riccati equation (ARE) for approximate dynamic programming (ADP) [20][21]. An identification NN was introduced for linear system identification to stabilize the convergence of the actor-critic networks for the consensus of a linear MAS further [22]. A policy iteration algorithm for the consensus of a nonlinear MAS was developed from ADP derived from an extra compensator [23]. Online RL to solve the optimal consensus problem was proposed for the consensus of a MAS with the second-order dynamics [24].

While RL has the potential of providing a consensus algorithm without explicit model knowledge, being sensitive to parameterization, it often experiences the convergence problem. The online RL in [24] applied to the consensus of a MAS with the second-order dynamics with unknown time-varying delay and disturbance was shown to converge very slowly in comparison to the consensus algorithm based on supervised deep learning [25]. To deal with the slow convergence, we propose an RL with the structure of the sliding mode control, which is called “slide RL”. The structure of the action follows the sliding mode control which consists of a linear part and a sliding part. The underlying motivation to introduce the structure of sliding mode control to RL is that the sliding mode control is robust to the uncertainties. It implies that RL and the sliding mode control can be considered as robust methods to the uncertainties. Thus, combining these two methods may accelerate the convergence of the RL. Simulation results with various system configurations show that the proposed slide RL achieves near model-based performance without model knowledge.

This paper is organized as follows. A system model for a MAS with the second-order dynamics and the consensus objectives are presented in section-II. The section-III provides the derivation of the slide RL and a description of the corresponding pseudo-code. In section-IV, the parameters of the sliding RL were determined first from simulations first. The performance of the slide RL was compared with the model-based algorithm and the existing state of art consensus RL algorithms. Some concluding remarks and the future research are presented in section-V.

Ⅱ. A System Model and Problem Formulation

In this paper, a consensus of a leader-follow multi-agent system with the second-order dynamics is considered. A corresponding system model can be given as

$$\ddot{x}_i(t) = u_i(t) + d_i(t) \tag{1}$$

where $\ddot{x}_i(t)$ is the second order derivative of the position $x_i(t)$ of the agent i, $u_i(t)$ is a control signal and $d_i(t)$ is disturbance from external sources such as wind. It is assumed that the system state information can be obtained without measurement error. If there is a measurement error, the state estimation algorithm needs to be developed further, which is beyond the scope of this research. It is also assumed that each agent transmits system information to a subset of agents which can be determined from the communication range or pre-plan to satisfy some goal of performance objective. A communication graph which consists of edges and nodes holds this information. The edge and the node represent the presence of communication and each agent respectively. Let A be the adjacency matrix of the corresponding communication graph and a_{ij} be the element located as the ith row and the jth column. a_{ij} has the value of 1 when the agent i receives information from the agent j, otherwise 0.

The model-based algorithm usually takes the control signal as a linear function of state difference in the absence of uncertainty. In the presence of uncertainties, the control signal for a consensus control can be represented as follows.

$$u_i(t) = f(\Omega_i(t)) \tag{2}$$

where $\Omega_i(t)$ is information available at time t for the agent i. $\Omega_i(t)$ can be written as

$$\begin{aligned}\Omega_i(t) = \{&\mathrm{x_j}(\mathrm{t}-\tau_{\mathrm{ij}}), \dot{\mathrm{x}}_{\mathrm{j}}(\mathrm{t}-\tau_{\mathrm{ij}}), \ddot{\mathrm{x}}_{\mathrm{j}}(\mathrm{t}-\tau_{\mathrm{ij}}) | \mathrm{j\,for\,a_{ij}} = 1\} \\ &\cup \{\mathrm{x_i}(\mathrm{t}), \dot{\mathrm{x}}_{\mathrm{i}}(\mathrm{t}), \ddot{\mathrm{x}}_{\mathrm{i}}(\mathrm{t})\}\end{aligned} \tag{3}$$

where τ_{ij} is a communication delay from agent j to agent i. It is assumed that the communication delay is unknown and time-varying. The time-varying unknown delay has an effect on the stabilization of the control and the convergence speed of the control.

The goal of the consensus control for a leader-followers MAS with the second-order dynamics is to synchronize the position of each follower agent to that of the leader agent. It can be represented as

$$q_{x,i}(t) = x_i(t) - x_0(t) \tag{4}$$

$$\lim_{t \to \infty} |q_{x,i}(t)| = 0, \text{ for } i = 1, 2, \cdots, N \tag{5}$$

where $q_i(t)$ is often called as a disagreement error. Even though the asymptotic convergence is considered, the convergence speed can be of practical interest. The following local error has been often used as a metric to measure the degree of consensus.

$$e_{x,i}(t) = \sum_{j=0}^{N} a_{ij}(x_i(t) - x_j(t)) \tag{6}$$

$$e_{v,i}(t) = \sum_{j=0}^{N} a_{ij}(\dot{x}_i(t) - \dot{x}_j(t)) \tag{7}$$

Ⅲ. Slide Reinforcement Learning

Sliding mode control has been researched to develop robust control in the presence of uncertainties. The basic principle in the sliding mode control is to pull up the state into the sliding surface where the state trajectory is designed to be independent of uncertainties. A model based RL[24] is exploited to be equipped with sliding mode. Since the sliding mode control is a model-based control, the model-based RL may provide a better fit to allow sliding mode in its structure. For the purpose of the implementation, The slide RL is developed for the MAS in a discrete time which is a sampled version of the continuous time system model with the sampling period T_s.

A discounted action-value function which measures the value of action for a given state can be defined as follows

$$V_{i,k} = \sum_{p=k}^{\infty} \gamma^{p-k} r_{i,k} = r_{i,k} + \gamma V_{i,k+1} \tag{8}$$

$$r_{i,k}=s_{i,k}^TP_is_{i,k}+e_{i,k}^TQ_ie_{i,k}+u_{i,k}^TR_iu_{i,k} \tag{9}$$

where $r_{i,k}$ is a reward signal of the agent i at time step k, γ is a discounting factor to determine how much future reward will be considered, and P_i, Q_i, and R_i are symmetric weight matrices with dimensions associated with $s_{i,k}$, $e_{i,k}$ and $u_{i,k}$ respectively. To exploit the sliding mode, $u_{i,k}$ can be structured in the following way [14].

$$u_{i,k}=u_{pre,i,k}-k_u sign(s_{i,k}) \tag{10}$$

where k_u is a weight controlling the effect of sliding mode, and $sign()$ is a sign function which takes a 1 if input is positive, 0 otherwise.

We take the same procedure to derive the optimal control policy as exploited in [24]. Let $V_{i,k}^*$ be the optimal value function which satisfies the coupled Hamilton-Jacobian equations. The dynamic programing equation for $V_{i,k}^*$ can be written as from (8)

$$V_{i,k}^*=r_{i,k}(u_{i,k}^*)+\gamma V_{i,k+1}^* \tag{11}$$

where $u_{i,k}^*$ is an optimal action of the agent i at time step k. From the first-order optimality condition, the optimal action can be expressed as

$$\frac{\partial r_{i,k}(u_{i,k}^*)}{\partial u_{i,k}^*}=-\gamma\frac{\partial V_{i,k+1}^*}{\partial u_{i,k}^*} \tag{12}$$

For a given $s_{i,k}$, (12) can be rewritten as

$$\frac{\partial r_{i,k}(u_{pre,i,k}^*)}{\partial u_{pre,i,k}^*}=-\gamma\frac{\partial V_{i,k+1}^*}{\partial u_{pre,i,k}^*} \tag{13}$$

With exploiting the derivation procedures and the fact that the discounted value function $V_{i,k+1}^*=z_{ci,i,k+1}^T\Phi z_{ci,i,k+1}$ can be represented as quadratic function, (13) can be arranged as

$$\frac{\partial r_{i,k}(u_{pre,i,k}^*)}{\partial u_{pre,i,k}^*}=-\frac{1}{2}R_i^{-1}(\sum_{j=0}^{N}a_{ij})\,G(\Phi+\Phi^T)z_{ci,i,k} \tag{14}$$

where $G=[T_s\ 0\ T_s 0]$ with the first 0 is a scalar and the last 0 is a row vector with the dimension depending on the number of neighbor agents, $z_{ci,i,k}=[s_{i,k}^T\ e_{i,k}^T\ u_{i,k}^T\ u_{i^-,k}^T]^T$, and $u_{i^-,k}$ is the concatenated neighbor actions at time step k. With limiting the actor network as a linear network, the proposed slide RL for the consensus of a leader-followers MAS with the second-order dynamics can be summarized as the figure-1 which is modified from the pseudo-code in [24]. It first initializes the parameters and variables. Then, it repeats the process from steps 1 to 6 at each time step. It generates the partial action from the actor network, with which the action is determined from combining the sliding part. Then, it calculates the action-state value and target partial action With the temporal difference error and the difference between the partial action and the target partial action, the critic network and the actor network are updated respectively.

```
0. Initialization
    Initialize elements of $W_i$ with $N(0,\sigma_1^2)$
    Initialize the nondiagonal elements of $\Phi$ with $N(0,\sigma_2^2)$
    Initialize the diagonal elements of $\Phi$ with $\sigma_2^2$
    Initialize $x_i$ with $N(0,\sigma_3^2)$ and $v_i$ with $N(0,\sigma_4^2)$
    Set learning rates $\kappa_{ai}$ and $\kappa_{ci}$
    Set $P_i$, $Q_i$, and $R_i$
1. $u_{pre,i,k}=W_iy_{a,i,k}$ where $y_{a,i,k}=[e_{x,i,k}^T\ e_{v,i,k}^T]^T$
2. $u_{i,k}=u_{pre,i,k}-k_usign(s_{i,k})$ where $s_{i,k}=ce_{x,i,k}+e_{v,i,k}$
3. $V_{i,k}=y_{c,i,k}^T\Phi_iy_{c,i,k}$ where $y_{c,i,k}=[s_{i,k}\ e_{x,i,k}^T\ e_{v,i,k}^T\ u_{i,k}^T\ u_{i^-,k}^T]^T$
4. $\hat{u}_{i,k}=-\frac{1}{2}R_i^{-1}\gamma(\sum_{j=0}^{N}a_{i,j})G(\Phi_i+\Phi_i^T)y_{c,i,k}$
5. $W_i=W_i-\kappa_{a,i}\xi_{a,i}y_{a,i,k}$ where $\xi_{a,i}=(u_{pre,i,k}-\hat{u}_{i,k})$
6. $\Phi_i=\Phi_i-\kappa_{c,i}\xi_{c,i}y_{c,i,k}$ where $\xi_{c,i}=(V_{i,k-1}-(r_{i,k-1}+\gamma V_{i,k}))$
```

그림 1. 지도자-추종자 다중 에이전트 시스템의 컨센서스를 위한 슬라이드 RL 구현 유사 코드

Fig. 1. Pseudo Code for the implementation of the slide RL for the consensus of a leader-followers MAS

Ⅳ. Experiment Results

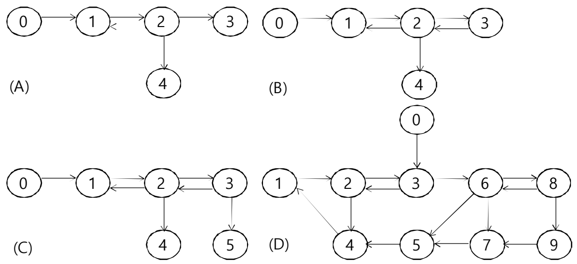


그림 2. 모의실험을 위한 통신 그래프

Fig. 2. Communication graphs used for simulation

In this section, the performance of the proposed slide RL is assessed with numerical simulations. To this end, the descriptions of the operations of a MAS will be provided with the parameterization and network structures of the neural networks for RL. The performance of the slide RL will be compared with several states of art methods.

Following the system configurations used in [25], 6 different communications graphs shown in the figure-2 were considered to assess the applicability of the proposed method to various environments. The number of nodes is 5 to 10 while the maximum number of neighbor nodes is 2 for the simplicity of the simulation. Delay, disturbance, and the dynamics of the leader agents were configured differently as presented in the table-1 and the table-2 where k and l are indices for agents. Different disturbance with the same envelope is applied to each agent. The delay at each communication link is different while the maximum delay is 0.5 seconds for every configuration.

Comparing state of art algorithms were configured as follows. The actor network for the modified TD3 with pretraining was configurated to have one hidden layer with 256 nodes and linear activation, and an output layer with 1 output node and hyper-tangent activation. The critic network was configured to have 3 hidden layers with a number of nodes, 32,256, and 256, and an output layer with one node and linear activation. The model-based RL and the model-based RL(p) were configured with following the description in [24][26]. Please refer to the details of the comparing algorithms in [24][25][26].

The slide RL was configured to have the same structure as the model-based RL except that the square of the sliding variable is added to the reward. Four important parameters were found to be critical in the convergence of the slide RL. Those are the initialization of the weight matrix W_i for generating action, the initialization of the weight matrix Φ_i for determining the action-state value, the learning rates, κ_a and κ_c for the actor network and the critic networks, and the sliding mode weight k_u. W_i and Φ_i are initialized with gaussian random variable with mean 0 and standard deviation, σ_1 and σ_2 respectively. Depending on these parameters, the state of an agent was observed to diverge with some different initializations for the same system configuration. To find a proper parameterization, different combinations of σ_1, σ_2, and κ_a were tried while k_u was fixed at 100, and κ_c was set as equal to κ_a. After having many different heuristic trials. σ_ϑ of 10^{-6}, σ_w of 10^{-6}, and κ_a of $5x10^{-10}$ were found to provide stable convergence with different initializations for each system configuration. Increasing κ_a to $5x10^{-9}$ with fixing σ_ϑ and σ_w as 10^{-6} resulted in the divergence for all system configurations. Increasing σ_w or σ_ϑ to 10^{-3} with fixing κ_a as $5x10^{-10}$ did not incur any divergence while it degrades the mean square error (MSE) performance slightly. These results may imply that the performance of slide RL has strong dependency on the learning rate while it has marginal dependency on the initializations of W_i and Φ_i. From these results, σ_1, σ_2, and κ_a and κ_c were set as $5x10^{-10}$ unless otherwise stated.

표 1. 각 시스템 구성에서 지도자 에이전트의 가속도와 왜란
Table 1. The acceleration of the leader agent and disturbance for each system configuration

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 2. 각 시스템 구성에서 통신 그래프와 지연
Table 2. The communication graph and delay for each system configuration

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	$0.5(1-0.5(1+\cos(t+(k+l)\pi/7))e^{-0.1(k+l)t})$

However, depending on k_u, the performance of the slide RL may vary. To examine the effect of k_u, the performance of the slide RL was evaluated for 4 different k_us. The table-3 shows the MSE with different k_us. It is observed that the MSE marginally increases with increasing k_u for all system configurations. A similar trend can be found on the mean square of disagreement (MSD) in the table-4 even though the effect of k_u on MSD is not as significant as on MSE. When k_s was set as 20 or 30, some divergences for the case 5 and case 6 were observed. This can be expected in the sense that k_s is required to be greater than some value to provide stable control performance in a sliding mode control. Since k_s of 40 provides the lowest MSE and MSD while saving the control efforts and providing stable convergence, k_s was set as 40 throughout this simulation unless otherwise stated.

After optimizing the parameters of the slide RL, the performance of the slide RL were compared with the existing state of art methods. They are model-based RL [24], pretrained model-based RL denoted as model-based RL(p)[26], pretrained modified TD3 denoted as modified TD3(p) [26]. The performances of the slide RL were shown for each case in figure-2. The slide RL was observed to provide comparable performance to the model-based algorithm with the best performance. The model-based RL without pretraining experienced divergence for some initializations for each case while the two pretrained RL algorithms provide consistent similar performance. The slide RL is also found to outperform the existing RLs in terms of the average MSE while the variance of the MSE of the slide RL is the smallest among considered RL algorithms for all cases.

표 3. k_u의 MSE에 대한 영향
Table 3. The effect of k_u on MSE

case\\k_u	40	60	80	100
1	0.00027	0.00070	0.00147	0.00256
2	0.00021	0.00056	0.00117	0.00203
3	0.00024	0.00064	0.00120	0.00223
4	0.00022	0.00056	0.00111	0.00167
5	0.00022	0.00056	0.00113	0.00184
6	0.00030	0.00071	0.00149	0.00210

표 4. k_u의 MSD에 대한 영향
Table 4. The effect of k_u on MSD

case\\k_u	40	60	80	100
1	0.00360	0.00415	0.00643	0.00698
2	0.00443	0.00473	0.00605	0.00714
3	0.01388	0.01484	0.01858	0.02076
4	0.01190	0.01370	0.01600	0.01933
5	0.01442	0.01372	0.01687	0.01982
6	0.01004	0.01174	0.01574	0.01623

To evaluate the performance quantitatively, the MSE and MSD was provided in the table-5 and the table-6. the performance of the model-based RL without pretraining was not included due to the occurrence of the divergence. However, it can be seen from the figure-3 that the model-based RL may provide reasonable performance when it does not diverge. The slide RL is shown to provide consistent MSE performance for all cases as the model-based RL does while the performance of two pretrained RL varies relatively significantly. The MSE of the slide RL is lower than the two pretrained RL by the order of 2 or so. The same trend can be observed for the MSD while the MSD is larger than MSE by the order of 1 or so for all algorithms due to the delay and disturbance. Nonetheless, the slide RL provides MSD small enough to be used in many practical system environments. To see how consistent performance the slide RL provided, the ratio of initialization of which MSE or MSD less than 0.1 is shown in the table-7. It can be observed that the slide RL and the model-based algorithm have MSE and MSD less than 0.1 for all initializations in all cases.

Some RL algorithms can provide good performance at convergence while other algorithms can converge quickly. Thus, it is important to assess the convergence characteristics of the slide RL. The figure-4 shows the MSE convergence of the slide RL for the case 1,3,4, and 6 which have a different number of agents. It can be observed that the model-based algorithm, the modified TD3(p), and the slide RL converges within 200 steps or so for all cases. The modified RL converges slowly since it learns how to converge at each step. The modified RL(p) converges faster than the modified RL since it already learned how to generate action. However, since it might not learn how to converge fast enough, its convergence speed is found to be slower than the fast ones.

표 5. 기존 우수 알고리즘과 비교한 제안 알고리즘의 MSE 성능 (x는 발산의 발생을 나타낸다.)
Table 5. MSE performance of the proposed algorithm in comparison with state of arts algorithms (x represents the occurrence of divergence)

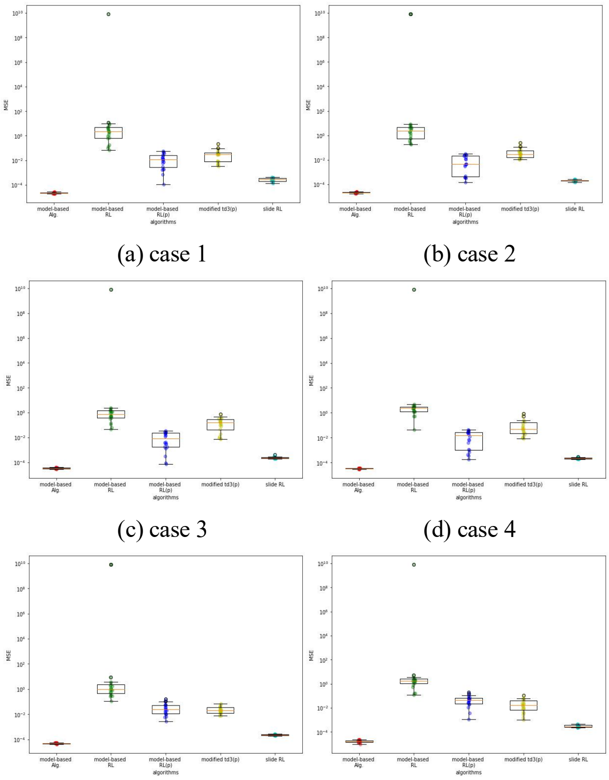
case	1	3	4	6
model-based Alg.	0.000022	0.000034	0.000033	0.000017
model-based RL	x	x	x	x
model-based RL(p)	0.016820	0.012400	0.016580	0.055990
modified TD3(p)	0.041020	0.188250	0.134127	0.026094
slide RL	0.000277	0.000242	0.000223	0.000302

표 6. 기존 우수 알고리즘과 비교한 제안 알고리즘의 MSD 성능 (x는 발산의 발생을 나타낸다.)
Table 6. MSD performance of the proposed algorithm in comparison with state of arts algorithms (x represents the occurrence of divergence)

case	1	3	4	6
model-based Alg.	0.000131	0.000641	0.000671	0.000142
model-based RL	x	x	x	x
model-based RL(p)	0.305236	0.785603	1.181799	6.431354
modified TD3(p)	0.308835	5.598264	4.766102	0.995609
slide RL	0.003605	0.013879	0.011895	0.010042

표 7. MSE와 MSD가 0.1보다 작은 값을 갖는 초기화의 비율
Table. 7. The ratio of initialization which has the MSE and MSD less than 0.1

case	1	3	4	6
model-based Alg.	1.00/1.00	1.00/1.00	1.00/1.00	1.00/1.00
model-based RL	0.05/0.00	0.10/0.00	0.05/0.00	0.00/0.00
model-based RL(p)	1.00/0.50	1.00/0.25	1.00/0.25	0.85/0.00
modified TD3(p)	0.90/0.30	0.40/0.15	0.65/0.05	0.95/0.20
slide RL	1.00/1.00	1.00/1.00	1.00/1.00	1.00/1.00



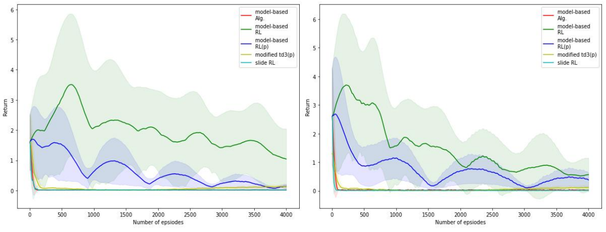
(a) case 1 (b) case 2

(c) case 3 (d) case 4

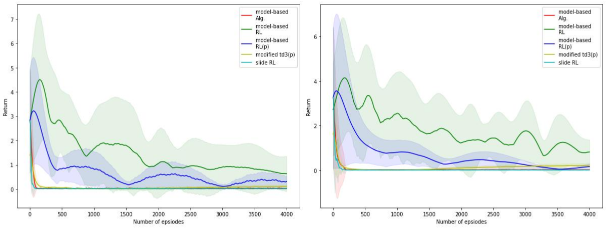
(e) case 5 (f) case 6

그림 3. 모의실험을 위한 통신 그래프 모의 실험 경우에 대한 MSE의 상자 그림 (x축의 레이블은 다음과 같다.model base alg., model-based RL, model-based RL(p), modified TD3(p), and slide RL. y축은 로그 스케일의 MSE)

Fig. 3. The boxplot of MSE for each case. (The x axis is labeled from the left to the right as follows. model base alg., model-based RL, model-based RL(p), modified TD3(p), and slide RL. y-axis is MSE in log scale)



(a) case 1 (b) case 3



(c) case 4 (d) case 6

그림 4 슬라이드 RL의 수렴 특성 (적색 : model-based algorithm, 녹색 : model-based RL, 청색 : model-based RL(p), 황색: modified TD3(p), and 청록색 : slide RL. x축은 에피소드의 수. y축은 리턴)

Fig. 4. Convergence characteristics of the slide RL. (red : model-based algorithm, green : model-based RL, blue : model-based RL(p), yellow : modified TD3(p), and cyan : slide RL. x-axis and y-axis are the number of episodes and return respectively)

Ⅴ. Conclusions

The state of art RL algorithms for the consensus of a MAS have been observed to converge slowly or have unstable convergence characteristics having a dependency on initialization. To overcome these issues, the slide RL which accelerates the convergence and improves the stability of the control through deriving the state variable on the sliding surface and making it robust to uncertainties. The proposed slide RL was shown to provide comparable MSE performance and convergence speed to those of the model-based algorithm while it outperforms the existing state of art algorithms.

Despite the superior performance of the slide RL, there remain many problems to be addressed further. while the slide RL was developed with the model-based RL, the conventional RL framework can be combined with the slide mode control. While the application of the slide RL to a control problem can be trivially straightforward, the application to the problem with a delayed reward or a vague system model is likely to necessitate the articulation of the surrogate of the sliding variable and surface. It will be also interesting to see how the sliding mode can be exploited to offline RL [27] to improve the performance.

Acknowledgement

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

References

[1] 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.

[2] M. Colombino, D. Groß and F. Dorfler, "Global phase and voltage synchronization for power inverters: A decentralized consensus-inspired approach," in *Proceeding of the 56th Annual Conference on Decision and Control (CDC), Melbourne, VIC, Australia*, pp. 5690-5695, Dec. 2017.

[3] H. Septanto, B. Riyanto-Trilaksono, A. Syaichu-Rohman and R. Eko-Poetro, "Consensus-based controllers for spacecraft attitude alignment: Simulation results," in *Proceeding of the 2nd International Conference on Instrumentation, Communications, Information Technology, and Biomedical Engineering, Bandung, Indonesia,* pp. 52-57, Dec. 2011.

[4] 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.

[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.

[6] A. T. Chin Loon and M. N. Mahyuddin, "Network server load balancing using consensus-based control algorithm," in *Proceeding of the IEEE Industrial Electronics and Applications Conference (IEACon), Kota Kinabalu, Malaysia,* pp. 291-296, Nov. 2016.

[7] R. Olfati-Saber, J. A. Fax and R. M. Murray, "Consensus and Cooperation in Networked Multi-Agent Systems," *Proceedings of the IEEE,* Vol. 95, No. 1, pp. 215-233, Jan. 2007.

[8] R. Olfati-Saber and R. M. Murray, "Consensus problems in networks of agents with switching topology and time-delays," *IEEE Trans. Autom. Control, V*ol. 49, No. 9, pp. 1520-1533, Sep. 2004.

[9] Y. Cao and W. Ren, "Optimal Linear-Consensus Algorithms: An LQR Perspective," *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, Vol. 40, No. 3, pp. 819-830, June 2010.

[10] D. H. Nguyen, "A sub-optimal consensus design for multi-agent systems based on hierarchical LQR," *Automatica,* Vol. 55, pp. 88-94, May 2015.

[11] A. Khalil, and J. Wang, "Stability and Time Delay Tolerance Analysis Approach for Networked Control Systems," *Mathematical Problems in Engineering,* Vol. 2015, pp.1-9, 2015.

[12] Q. Zhang, Y. Niu, L. Wang, L. Shen, and H. Zhu, "Average consensus seeking of high-order continuous-time multi-agent systems with multiple time-varying communication delays," *Int. J. Control Autom. Syst.,* Vol. 9, pp. 12090-1218, Dec. 2011.

[13] Y.-J. Sun, G.-L. Zhang, and J. Zeng, "Consensus analysis for a class of heterogeneous multiagent systems with time delay based on frequency domain method," *Math. Problems Eng.,* Vol. 2014, pp. 1-7, Sep. 2014.

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

[15] D. Zhang, L. Liu, and G. Feng, "Consensus of Heterogeneous Linear Multiagent Systems Subject to Aperiodic Sampled-Data and DoS Attack," *IEEE Transactions on Cybernetics*, Vol. 49, No. 4, pp. 1501 - 1511, April 2019.

[16] O. Vinyals, I. Babuschkin, W. M. Czarnecki, et al., "Grandmaster level in StarCraft II using multi-agent reinforcement learning," *Nature,* Vol. 575, pp. 350-354, 2019.

[17] J. B. Kim, H. K. Lim, C. M. Kim, M. S. Kim, Y. G. Hong and Y. H. Han, "Imitation Reinforcement Learning-Based Remote Rotary Inverted Pendulum Control in OpenFlow Network," *IEEE Access*, Vol. 7, pp. 36682-36690, 2019.

[18] S. Gu, E. Holly, T. Lillicrap and S. Levine, "Deep reinforcement learning for robotic manipulation with asynchronous off-policy updates," in *Proceeding of IEEE International Conference on Robotics and Automation (ICRA), Singapore*, pp. 3389-3396, July 2017.

[19] 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.

[20] 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.

[21] 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), Singapore,* pp. 606-611, Oct. 2020.

[22] Y. Liu, T. Li, Q. Shan, R. Yu, Y. Wu, 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.

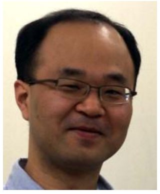
[23] J. Zhang, H. Zhang and T. Feng, “Distributed Optimal Consensus Control for Nonlinear Multiagent System With Unknown Dynamic,” *IEEE Transactions on Neural Networks and Learning Systems*, Vol. 29, No. 8, pp. 3339-3348, Aug. 2018.

[24] 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.

[25] 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.

[26] J. Yang, “Reinforcement Learning for the Consensus of Multi-agents with Unknown Time Varying Delays,” *Journal of Digital Contents Society,* Vol. 23, No. 7, pp. 1,277 – 1,287, July 2022.

[27] M. Li, X. Gao, Y. Wen, J. Si and H. H. Huang, “Offline Policy Iteration Based Reinforcement Learning Controller for Online Robotic Knee Prosthesis Parameter Tuning,” in *Proceeding of the International Conference on Robotics and Automation (ICRA), Montreal, QC, Canada,* pp. 2831-2837, May 2019.



양 장 훈 (Janghoon Yang)
2006년 8월 : University of Southern California, Department of Electrical Engineering (공학박사)
2010년3월 ~ 현재 : 서울미디어대학원대학교 인공지능 응용소프트웨어학과 부교수
※관심분야 : 제어, 인공지능, 통신, 감성, 콘텐츠