Journal of Machine Learning and Information Security
https://www.sciltp.com/journals/jmlis

Article

Data-Based Optimal Couple-Group Consensus Control for Heterogeneous Multi-Agent Systems via Policy Gradient Reinforcement Learning

Jun Li [1], Xiaoyu Pei [2] and Lianghao Ji [2,*]

[1] School of Economics and Management, Chongqing University of Posts and Telecommunications, Chongqing 400065, China
[2] Chongqing Key Laboratory of Image Cognition, School of Computer Science and Technology, Chongqing University of Posts and Telecommunications, Chongqing 400065, China
* Correspondence: jun_li2023@163.com

How To Cite: Li, J.; Pei, X.; Ji, L. Data-Based Optimal Couple-Group Consensus Control for Heterogeneous Multi-Agent Systems via Policy Gradient Reinforcement Learning. *Journal of Machine Learning and Information Security* **2026**, *2*(1), 1. https://doi.org/10.53941/jmlis.2026.100001

Received: 30 August 2025
Revised: 17 December 2025
Accepted: 5 January 2026
Published: 13 January 2026

Abstract: This paper investigates the optimal couple-group consensus control (OCGCC) for heterogeneous multi-agent systems (HeMASs) with completely unknown dynamics. The agents in HeMASs are divided into two groups according to order differences. Meanwhile, heterogeneous systems are transformed into homogeneous ones by adding virtual velocities. Then, a novel data-driven distributed control protocol for HeMASs is proposed based on policy gradient reinforcement learning (RL). The proposed algorithm is implemented asynchronously, and is specifically designed to address the issue of computational imbalance caused by individual differences among participants. It achieves this by constructing an actor-critic (AC) framework. The system's learning efficacy is optimized using offline data sets. The convergence and stability are ensured by applying functional analysis and the Lyapunov stability theory. Finally, the effectiveness of the proposed algorithm is confirmed by various simulation examples.

Keywords: consensus control; reinforcement learning; policy gradient; heterogeneous multi-agent systems

1. Introduction

In recent years, distributed control in multi-agent systems (MASs) has garnered significant attention due to its wide-ranging applications across diverse fields, such as formation control [1], smart grids [2], intelligent transportation systems [3–5], and event-triggered control [6]. Among the core challenges of MASs, the consensus control problem stands out as pivotal, with its ultimate objective being to enable all agents to reach agreement on a state of interest through local interactions. However, in practical scenarios, the resources required to achieve control objectives are often limited. Consequently, the optimal consensus control (OCC) problem has emerged as a widely studied topic [7], that is, agents not only achieve consensus but also a minimize performance index such as energy consumption or control cost.

The OCC problem typically requires solving coupled Hamilton-Jacobi-Bellman (HJB) equations. Nevertheless, the non-linearity of HJB equations and the interdependence among agents make analytical solutions generally unavailable. Furthermore, most existing model-based control methods rely on the exact dynamics of the MASs, which are often difficult to obtain for large-scale systems. The emergence of RL, as a data-driven method for learning through agent-environment interaction, has provided a highly promising avenue to circumvent these obstacles. By leveraging post-exploration data, RL can learn the optimal control law without the system dynamics for OCC problems [8–11]. At present, many RL-based approaches have been developed, such as policy interaction techniques [12], neural network-based methods [13], adaptive predictive control [14], adaptive cruise control [15], and Q-learning-based methods [16,17]. Experience replay [18] and fuzzy logic systems [19] have further been employed to enhance data efficiency and robustness. It is worth noting that, compared with value iteration [20], the policy gradient (PG) method has more advantages in handling the continuous action space and ensuring stability [20].

Recent studies have addressed practical issues such as asynchronous updates in homogeneous MASs [21]. These comprehensive advancements strongly demonstrate that RL is a reasonable and powerful framework for solving the OCC problems in MASs. While striving for optimality, the research scope of MASs has expanded from homogeneous systems where all agents have the same dynamic characteristics to HeMASs with different dynamic characteristics [22]. Although significant progress has been made in the optimization control of HeMASs [23–25], most studies have focused on agents with the same state dimensions. The problem of mixed-order HeMASs, which involve agents with fundamentally different dynamic orders, has received relatively less attention in the data-driven RL community. Existing research on mixed-order systems typically relies on frequency-domain analysis [26,27] and introduces virtual velocities to unify the system [12,13,16,28]. Although these methods simplify the analysis, may not fully capture the inherent heterogeneity and can introduce additional learning costs.

In addition, the demand for multi-task processing capabilities has led to the emergence of group consensus issues. In this case, the system is divided into multiple subgroups, and the control objective is to achieve consensus within each subgroup while allowing for differences between them [29,30]. One particularly notable variant involves cooperative-competitive interactions, where agents within the same subgroup cooperate with each other, while agents from different subgroups compete with each other [31–34]. This framework is closely related to the MASs with mixed-order state spaces, as different subgroups are typically composed of agents with different dynamic characteristics. However, the integration of cooperative-competitive interactions with mixed-order group consensus remains underexplored, especially from an optimal control perspective using RL. The core challenge lies in how the RL framework can effectively manage the complex information exchange among agents from different subgroups.

Apart from the algorithmic challenges brought about by system heterogeneity, another crucial practical consideration is the variation in computational capabilities among agents. In actual deployment, agents may have varying processing capabilities, resulting in an uneven distribution of strategy update times. A synchronous learning algorithm that requires all agents to update simultaneously would be forced to wait for the slowest agent, thereby severely impeding the overall learning efficiency. Although asynchronous RL algorithms have been proposed for homogeneous MASs [21], in mixed-order MASs, this problem becomes even more severe because the differences in dynamics may further exacerbate this asynchrony.

Based on the aforementioned related studies, this work employs a PG algorithm to explore the OCGCC problem for a class of HeMASs within the leader-follower framework. In summary, the main contributions are as follows:

(1) Taking into account the differences in computational performance among agents, a novel asynchronous RL method is proposed, which resolves the issue of inconsistent strategy-update speeds among agents. The proposed algorithm addresses the asynchronous issues caused by mixed-order dynamics by designing an asynchronous update mechanism. It is capable of handling interaction complexity goes beyond merely considering computational-capability differences, as discussed in [21,24]. By combining offline data with experience replay, and proving the asynchronous convergence of mixed-order systems in competition-cooperation topologies within the Lyapunov framework, it expands the theoretical analysis in [21,24,27,35].

(2) A distributed optimal control law is derived to balance the information exploitation and the data exploration, thereby optimizing decision-making processes in MASs. Additionally, the experience replay scheme is introduced to break the temporal correlation of consecutive data samples [18,24]. This involves randomly sampling past experiences to enhance learning stability and significantly improve the overall data utilization efficiency.

(3) The cooperation-competition mechanism is considered to address the communication and coordination challenges in the OCGCC problem [11,20,26]. In the absence of model-accurate information, optimal group consensus can still be achieved in both location and velocity states.

The symbols and explanations of the variables in this paper are shown in Table 1. The succeeding sections of this article are organized in the following manner. Section 2 contains the preliminary information. Section 3 provides the convergence analysis of the asynchronous RL algorithm. Section 4 presents the actor-critic (AC) network framework required to implement the algorithm. The efficiency of the algorithm is verified through the numerous simulation examples in Section 5. Finally, Section 6 concludes this article.

Table 1. Variable Symbol Interpretation.

Symbols	Meanings
$\mathfrak{G}$	Communication topology graph
$\mathfrak{V}$	Set of nodes
$\mathfrak{E}$	Set of edges
$\mathfrak{A}$	Weighted adjacency matrix
$\mathfrak{L}$	Laplacian matrix
N	Number of agents
β_1	Set of second-order agents
β_2	Set of first-order agents
$x_i(t)$	Position state of the i-th agent
$v_i(t)$	Velocity state of the i-th agent
$u_i(t)$	Control input of the i-th agent
$w_i(t)$	Estimated velocity of first-order agents
$\mathscr{A}, \mathscr{B}, \mathscr{C}$	System matrices
$\phi_i, \tilde{\phi}_i$	Input matrices
$\varsigma_i(t)$	Augmented state vector
Ξ	Augmented system matrix
$\xi_i, \bar{\xi}_i$	Augmented input matrices
$\epsilon_i(t)$	Tracking error of the i-th agent
$E(t)$	Global consensus error
a_{ij}	Element of adjacency matrix
b_i	Pinning gain
s_{ij}	Cooperation-competition strength factor
$\mathfrak{J}i$	Performance function of the i-th agent
$\mathscr{V}_i$	State-value function of the i-th agent
$\mathscr{Q}_i$	Q-function of the i-th agent
Q_{ii}, R_{ii}, R_{ij}	Performance index weight matrices of the i-th agent
$\mathscr{H}_i$	Hamilton function of the i-th agent
ρ_i	Learning rate
W_{ci}	Weight matrix of critic network
W_{ai}	Weight matrix of actor network
δ_{ci}, δ_{ai}	Activation functions
$h_{ci}(t), h_{ai}(t)$	Input vectors of neural networks
β_c, β_a	Learning rates of neural networks

2. Preliminaries

2.1. Algebraic Graph Theory

The directed communication of MASs is denoted by $\mathfrak{G} = (\mathfrak{V}, \mathfrak{E}, \mathfrak{A})$. This graph $\mathfrak{G}$ consists of a node set $\mathfrak{V} = \{v_1, v_2, \ldots, v_N\}$ with N agents, an edge set $\mathfrak{E} = \{e_{ij} = (v_i, v_j) \in \mathfrak{V} \times \mathfrak{V} \mid i, j \in \mathcal{I}\}$, and a weighted adjacency matrix $\mathfrak{A} = [a_{ij} \mid i, j \in \mathcal{I}]$ where $\mathcal{I} = \{1, 2, \ldots, N\}$ represents the finite set of node indices, $e_{ij} = (v_i, v_j)$ denotes the directed edge from node i to node j. And the information from node i can be received by node j only if $a_{ij} \neq 0$, indicating that the pair (v_i, v_j) is an element of $\mathfrak{E}$. The connection between nodes in graph $\mathfrak{G}$ is described by the weighted adjacency matrix $\mathfrak{A}$ with elements $\{-1, 0, 1\}$. When $a_{ij} > 0$, it indicates a cooperative link between nodes i and j. On the other hand, $a_{ij} < 0$ signifies a competitive interaction. Nodes i and j do not interact if $a_{ij} = 0$.

The in-degree matrix $\mathfrak{D}$ is a diagonal matrix where the diagonal components d_i are calculated as $d_i = \sum_{j \in \mathcal{I}} a_{ij}$. The Laplacian matrix of $\mathfrak{G}$ is defined as $\mathfrak{L} = \mathfrak{D} - \mathfrak{A}$. Besides, there exists at least one directed path from the root node to every other node.

2.2. Problem Formulation

Consider a discrete-time HeMAS consisting of N followers and one leader. The dynamics of the i-th follower can be described as

$$\begin{cases} \begin{cases} x_i(t+1) = \mathscr{A}x_i(t) + \mathscr{B}v_i(t) \\ v_i(t+1) = \mathscr{C}v_i(t) + \phi_i u_i(t) \end{cases}, & i \in \beta_1 \\ x_i(t+1) = \mathscr{A}x_i(t) + \widehat{\phi}_i u_i(t), & i \in \beta_2 \end{cases}, \tag{1}$$

where $\beta_1 = \{1, \ldots, q\}$ denotes the set of second-order agents, and $\beta_2 = \{q+1, \ldots, N\}$ represents the first-order agents. $\sigma = \beta_1 \cup \beta_2$ and $\beta_1 \cap \beta_2 = \varnothing$. The control input, velocity state, and position state are represented by $u_i(t) \in \mathbb{R}^{n_i}, v_i(t), x_i(t) \in \mathbb{R}^p$, respectively. The matrices$\mathscr{A}, \mathscr{B}, \mathscr{C} \in \mathbb{R}^{p\times p}$ and the matrix $\phi_i, \widehat{\phi}_i \in \mathbb{R}^{p\times n_i}$ are unknown but constant matrices, and they serve as system and input matrices, respectively. In addition, the leader dynamics, which are not influenced by the followers, are given by

$$\begin{cases} x_0(t+1) = \mathscr{A}x_0(t) + \mathscr{B}v_0(t) \\ v_0(t+1) = \mathscr{C}v_0(t) \end{cases}, \tag{2}$$

in which $x_0(t), v_0(t) \in \mathbb{R}^p$ stand for position state and velocity state. Additionally, the OCGCC problem is formulated by introducing several definitions and assumptions.

Assumption 1. *The digraph $\mathfrak{G}$ is divided into two subgroups, which consist of the first-order agents and second-order agents, respectively. Moreover, the agents in mixed-order HeMASs (1) and (2) follow a cooperative–competitive interaction rule, i.e., agents within the same subgroup cooperate with each other, whereas agents from different subgroups compete with each other.*

Assumption 1 implies that the objective of group consensus control is to ensure that distributed agents within the same subgroup reach an identical state, while agents belonging to different subgroups converge to different states. In addition, the works [34,35] describe the cooperative-competitive connection among the agents in HeMASs as $(y_j - y_i)$ and $(y_j + y_i)$.

Definition 1. ***(Group Consensus of HeMASs):*** *The group consensus problem of HeMASs (1) and (2) can be asymptotically solved provided that the following conditions are met:*

$$(i) \begin{cases} \lim_{t\to\infty}\|x_i(t) - x_j(t)\| = 0, & \text{if } \eta_i = \eta_j \\ \lim_{t\to\infty}\|x_i(t) - x_j(t)\| \neq 0, & \text{if } \eta_i \neq \eta_j \\ \lim_{t\to\infty}\|v_i(t) - v_j(t)\| = 0, & \text{if } \eta_i = \eta_j \\ \lim_{t\to\infty}\|v_i(t) - v_j(t)\| \neq 0, & \text{if } \eta_i \neq \eta_j \end{cases},$$

$$(ii) \begin{cases} \lim_{t\to\infty}\|x_i(t) - x_0(t)\| = 0 \\ \lim_{t\to\infty}\|v_i(t) - v_0(t)\| = 0, \end{cases} \text{, if } \eta_i = \eta_0$$

where $\eta_p = \eta_q$ indicates that agent p and agent q belong to the same subgroup, while the opposite case means that p and q belong to different subgroups. In other words, agents within the same subgroup reach identical states, whereas agents in different subgroups converge to different states.

For analytical convenience, an estimated velocity [28] is introduced to convert first-order agents to second-order ones. Then the system dynamics in (1) can be rewritten as

$$\begin{cases} x_i(t+1) = \mathscr{A}x_i(t) + \mathscr{B}v_i(t) \\ v_i(t+1) = \mathscr{C}v_i(t) + \phi_i u_i(t), & i \in \beta_1 \\ \begin{cases} x_i(t+1) = \mathscr{A}x_i(t) + \mathscr{B}w_i(t) + \widehat{\phi}_i u_{i1}(t) \\ w_i(t+1) = \mathscr{B}w_i(t) + \widehat{\phi}_i u_{i2}(t) \end{cases}, & i \in \beta_2 \end{cases}, \tag{3}$$

where $w_i(t) \in \mathbb{R}^p$ represents the first-order agent' estimated velocity, $u_{i1}(t), u_{i2}(t) \in \mathbb{R}^{n_i}$ and $u_i(t) = u_{i1}(t) + u_{i2}(t)$.

Remark 1. *Many works [28,36] assume that agents' states in various subgroups have the same dimension. In contrast, the state-dimension is unequal in our study, and this issue is addressed by using estimated velocity. According to the description and setting of virtual velocity in the article [12,13,16], the virtual velocity is introduced*

to the first-order agents to convert them into second-order agents, making the dynamic equations of all agents consistent in form and thus converting the heterogeneous system into a homogeneous one for subsequent analysis.

By combining Equations (2) and (3), we can obtain

$$\begin{cases} \begin{cases} \varsigma_i(t+1) = \Xi\varsigma_i(t) + \xi_i u_i(t), & i \in \beta_1 \\ \varsigma_i(t+1) = \Xi\varsigma_i(t) + \widetilde{\xi}_i \widetilde{u}_i(t), & i \in \beta_2 \end{cases} , \\ \varsigma_0(t+1) = \Xi\varsigma_0(t) \end{cases} \tag{4}$$

where $\varsigma_i(t) = (x_i(t); v_i(t))$ for $i \in \beta_1, \varsigma_i(t) = (x_i(t); w_i(t))$ for $i \in \beta_2, \varsigma_0(t) = (x_0(t); v_0(t)), \Xi = \begin{pmatrix} \mathscr{A} & \mathscr{B} \\ \mathbf{0} & \mathscr{C} \end{pmatrix}, \xi_i = (\mathbf{0}; \phi_i) \in \mathbb{R}^{2p \times n_i}, \widetilde{\xi}_i = \text{diag}\left\{\hat{\phi}_i, \hat{\phi}_i\right\} \in \mathbb{R}^{2p \times 2n_i}, \widetilde{u}_i(t) = (u_{i1}(t); u_{i2}(t))$. Above all, to achieve group consensus of HeMASs, the tracking error of every agent is explicitly described as

$$\epsilon_i(t) = \sum_{j \in S_i} a_{ij}(s_{ij}\varsigma_j(t) - \varsigma_i(t)) + \sum_{j \in D_i} a_{ij}(s_{ij}\varsigma_j(t) - \varsigma_i(t)) + b_i(\varsigma_0(t) - \varsigma_i(t)). \tag{5}$$

The set S_i represents the neighboring nodes within the same subgroup, while D_i represents the corresponding nodes in a separate subgroup for agent i. $N_i = S_i \cup D_i, \varnothing = S_i \cap D_i$. Based on Assumption 1, there are $s_{ij} > 0 (j \in S_i)$ and $s_{ij} < 0 (j \in D_i)$. The constant b_i, referred to as pinning gains, are non-negative. If the information can be directly received from the leader by agent i, then $b_i > 0$; otherwise, $b_i = 0$. Define the global consensus error as

$$E(t) = (e_1(t); \ldots; e_m(t); e_{m+1}(t); \ldots; e_N(t)), \tag{6}$$

where $e_i(t)$ denotes the local consensus error of the i-th agent. The agents indexed by $\{1, 2, \ldots, m\}$ are second-order agents, and the remaining ones are first-order agents. Based on the above Assumptions 1 and 2, as well as the detailed theoretical derivation in [22], the global consensus error $E(t)$ can converge asymptotically for the HeMASs in (5).

2.3. Optimal Consensus Control via Nash Equilibrium

The local performance function of agent i on an infinite horizon can be expressed below for the purpose of addressing the issue of distributed OCC:

$$\mathfrak{J}_i(\epsilon_i(t), u_i(t), u_j(t)) = \sum_{\lambda=t}^{\infty} r(\epsilon_i(\lambda), u_i(\lambda), u_j(\lambda)), \tag{7}$$

where $u_j(t)$ are the control inputs of the neighbors j. $r(\epsilon_i(\lambda), u_i(\lambda), u_j(\lambda)) = \epsilon_i^T(\lambda)Q_{ii}\epsilon_i(\lambda) + u_i^T(\lambda)R_{ii}u_i(\lambda) + \sum u_j^T(\lambda)R_{ij}u_j(\lambda)$, in which Q_{ii}, R_{ii} and R_{ij} are positive symmetric matrices. Thus the local state-value function of agent i can be generated as

$$\mathscr{V}_i(\epsilon_i(t)) = \sum_{\lambda=t}^{\infty} r(\epsilon_i(\lambda), u_i(\epsilon_i(\lambda)), u_j(\epsilon_j(\lambda))), \tag{8}$$

and the following Bellman equation can be obtained by

$$\mathscr{V}_i(\epsilon_i(t)) = r(\epsilon_i(t), u_i(\epsilon_i(t)), u_j(\epsilon_j(t))) + \mathscr{V}_i(\epsilon_i(t+1)). \tag{9}$$

According to (8) and (9), the Hamilton function of the agent i is given by

$$\mathscr{H}_i(\epsilon_i(t), u_i(\epsilon_i(t)), \mathscr{V}) = r(\epsilon_i(t), u_i(\epsilon_i(t)), u_j(\epsilon_j(t))) + \mathscr{V}_i(\epsilon_i(t+1)) - \mathscr{V}_i(\epsilon_i(t)). \tag{10}$$

For the MASs described by (1) and the performance function given by (7), a control strategy is considered admissible if and only if it can stabilize the error system specified in (5) and ensure that the state-value function in (8) remains finite for all $i \in \mathbb{N}$.

Assumption 2. *The sets* $\left(\Xi, \xi_i, \widetilde{\xi}_i\right)$ *in* (4) *are controllable while* $\left(\Xi, Q_i^{1/2}\right)$ *are observable.*

In Assumption 2, the controllability of the system ensures the existence of a control input that can drive the system state from any initial point to any target point. This is the fundamental prerequisite for the existence of

an optimal control law. The observability of the system (equivalent to detectability here) ensures that the state weight matrix Q_{ii} in the performance index can fully reflect the internal state of the system. This ensures that the value function $\mathscr{V}_i$ is a positive definite function w.r.t. the system state, which can thus serve as a Lyapunov candidate function for stability analysis. In practice, we do not need to verify in advance whether these matrices satisfy Assumption 2. This assumption is a theoretical condition to ensure the control law obtained is optimal and stable after the algorithm converges.

The policy-based iterative method outperforms the value iteration method in evaluating the impact of the action space on the value. Introducing Q-function is crucial for quickly assessing the influence of action modifications on value. Therefore, this strategy provides better stability assurance than using only value iteration. The PG-based OCGCC method is facilitated by the Q-function considered as

$$\begin{aligned}\mathscr{Q}_i(\epsilon_i(t),u) &= \mathscr{Q}(\epsilon_i(t),u,u_j(\epsilon_j(t))) \\ &= r(\epsilon_i(t),u,u_j(\epsilon_j(t))) + \sum_{\lambda=t+1}^{\infty} r(\epsilon_i(\lambda),u_i(\epsilon_i(\lambda)),u_j(\epsilon_j(\lambda))).\end{aligned} \tag{11}$$

By integrating the local state-value function (8), it is deduced that

$$\begin{aligned}\mathscr{Q}_i(\epsilon_i(t),u) =& r(\epsilon_i(t),u,u_j(\epsilon_j(t))) + \mathscr{Q}_i(\epsilon_i(t+1),u_i(\epsilon_i(t+1))) \\ =& r(\epsilon_i(t),u,u_j(\epsilon_j(t))) + \mathscr{V}_i(\epsilon_i(t+1)).\end{aligned} \tag{12}$$

In accordance with the Bellman optimality principle, the following HJB equation is satisfied by a state-value function that is optimal:

$$\mathscr{V}_i^*(\epsilon_i(t)) = \min_{u_i(t)}\{r(\epsilon_i(t),u_i(t),u_j(\epsilon_j(t))) + \mathscr{V}_i^*(\epsilon_i(t+1))\}, \tag{13}$$

which provides the most suitable local Q-function for agent i in the following that

$$\mathscr{Q}_i^*(\epsilon_i(t),u) = \min_u r\big(\epsilon_i(t),u,u_j^*(\epsilon_j(t))\big) + \mathscr{Q}_i^*(\epsilon_i(t+1),u_i(\epsilon_i(t+1))). \tag{14}$$

In accordance with (13) and (14), the local OCC policy $u_i^*(\cdot)$ is written as

$$u_i^*(\epsilon_i(t)) = \arg\min_{u_i(t)} \mathscr{V}_i^*(\epsilon_i(t)) = \arg\min_u \mathscr{Q}_i^*(\epsilon_i(t),u), \tag{15}$$

where the corresponding control policy u is the best. We can conclude that $u_i^*(t) = \frac{1}{2}(d_i+b_i)R_{ii}^{-1}\mathbf{F}_i^\top \frac{\partial \mathscr{V}_i^*(\epsilon_i(t+1))}{\partial \epsilon_i(t+1)}$, where $i \in \beta_1$, $\mathbf{F} = \xi$, else $\mathbf{F} = \widetilde{\xi}$. Detailed theoretical derivation can be seen in [22].

Each agent i needs to acquire the following two types of information from its direct neighbors when performing policy evaluation and policy improvement steps. The neighborhood state information includes the position state $x_j(t)$, and velocity state $v_j(t)$ for second-order agents or estimated velocity $w_j(t)$ for first-order agents, which is used to calculate the local tracking error $\epsilon_i(t)$. Meanwhile, the neighbor's control policy $u_{j,p}(\epsilon_j(t))$ at the iteration step p is used to construct the Q-function for policy evaluation.

Remark 2. *Compared to the value iteration method, the policy iteration method starts from an initial acceptable policy to ensure the stability of the system [20]. This suggests that it is more effective in ensuring system admissibility and stability when compared to the iterative technique. Furthermore, by optimizing the agent using the action state value function, it gains experience with each action it takes, resulting in better real-time performance.*

Remark 3. *The adoption of virtual velocity reflects a strategic trade-off in the control of mixed-order HeMASs. This method can derive a unified, data-based control rule and provide a relatively simple stability analysis for the entire heterogeneous system. The proposed asynchronous RL architecture, combined with the experience replay technique, aims to reduce the associated computational costs. Future research will investigate alternative formulations that can achieve a similar homogenization effect while maintaining a lower dimensionality.*

3. Main Results

The section presents an analysis of the stability and convergence of a data-based asynchronous PG-based optimum algorithm.

3.1. Data-Based Asynchronous PG-Based Algorithm

Variations in computational capacity across agents in reality lead to varying iteration times and asynchrony during policy updating. Therefore, we extend the traditional PG algorithm to an asynchronous version. Compared to the traditional algorithm that suffers from synchronization delays [21], the proposed method is a gradient-based asynchronous policy algorithm, which is data-driven and consists of two main learning stages. In each iteration step, the policy is evaluated using the Bellman optimality equation, and the policy is enhanced based on the action gradient determined by the Q function. Algorithm 1 describes the detailed complexity.

Inspired by [21,35], each agent i is assigned a fixed period T_i, and the asynchronous policy update mechanism selects $M_p = \{i \in \mathbb{N} \mid p \bmod T_i = 0\}$ based on this period. Agents with stronger computing capabilities have smaller T_i values, and thus have a higher update frequency, while agents with limited resources have larger T_i values, and thus have a lower update frequency, thereby avoiding synchronization issues in HeMASs. The capabilities of each agent are determined by actual factors, such as its bandwidth and the efficiency of network transmission.

Remark 4. *In Algorithm 1, the computational speed of each agent is taken into account, which makes the algorithm more consistent with the real scenario than the outcomes reported in [27]. During each iteration, only a portion of the nodes participate in the update. This means that agents with stronger computing capabilities may pause their processing to accommodate slower agents, thus avoiding the problems caused by asynchronous strategy updates.*

3.2. Stability Analysis

To ensure the stability of the HeMASs using the suggested data-driven asynchronous PG-based Algorithm 1, it is necessary to incorporate an additional lemma.

Lemma 1. *Suppose that Assumption 1 holds. Assume that there exists the initial admissible control laws* $(u_{1,0}(\cdot), u_{2,0}(\cdot), \ldots, u_{n,0}(\cdot))$*, which are improved by Algorithm 1, and the corresponding Q-function is calculated according to Equations (20) and (23). If there exists* $\underline{\rho}_i < \rho_i < \bar{\rho}_i$*, then it holds that*

$$\mathscr{H}_i(\epsilon_i(t), u_{i,p+1}(\epsilon_i(t)), \mathscr{V}_{i,p}) = r(\epsilon_i(t), u_{i,p+1}(\epsilon_i(t)), u_{j,p+1}(\epsilon_{j,p}(t))) + \mathscr{V}_{i,p}(\epsilon_i(t+1)) - \mathscr{V}_{i,p}(\epsilon_i(t)) \leq 0, \tag{16}$$

where the specifics of $\underline{\rho}_i$ and $\bar{\rho}_i$ can be found in references [21,37].

Theorem 1. *Let Assumptions 1 and 2 hold true. Then the tracking error* $\epsilon_i(t)$ *is asymptotically stable. The HeMASs (1) will achieve group consensus by Algorithm 1, if the learning rate* $\underline{\rho}_i < \rho_i < \bar{\rho}_i$*.*

Proof. Suppose the initial policies $u_{1,0}(\cdot)$, $u_{2,0}(\cdot)$, …, $u_{n,0}(\cdot)$ are admissible. Then the coupled local policies $u_{1,p}(\cdot)$, $u_{2,p}(\cdot)$, …, $u_{n,p}(\cdot)$ are also admissible at $i = p$. Subsequently, it remains to verify the admissibility of the updated local policy $u_{i,p+1}$ and the policies of its neighboring agents $u_{j,p+1}$. □

The Lyapunov function can be chosen as the state-value function:

$$\mathscr{V}_{i,p}(\epsilon_i(t)) = r(\epsilon_i(t), u_{i,p+1}(\epsilon_i(t)), u_{j,p+1}(\epsilon_{j,p}(t))) + \mathscr{V}_{i,p}(\epsilon_i(t+1)). \tag{17}$$

Based on the error trajectory (5), the difference of $\mathscr{V}_{i,p}(\epsilon_i(t))$ can be calculated, and subsequently, we can obtain

$$\begin{aligned}
\mathscr{V}_{i,p}(\epsilon_i(t)) =& \mathscr{V}_{i,p}(\epsilon_i(t+1)) - \mathscr{V}_{i,p}(\epsilon_i(t)) \\
=& \mathscr{V}_{i,p}(\epsilon_i(t+1)) - \mathscr{V}_{i,p}(\epsilon_i(t)) + r(\epsilon_i(t), u_{i,p+1}(\epsilon_i(t)), u_{j,p+1}(\epsilon_{j,p}(t))) \\
& - r(\epsilon_i(t), u_{i,p+1}(\epsilon_i(t)), u_{j,p+1}(\epsilon_{j,p}(t))) \\
=& \mathscr{H}_i(\epsilon_i(t), u_{i,p+1}(\epsilon_i(t)), \mathscr{V}_{i,p}) - r(\epsilon_i(t), u_{i,p+1}(\epsilon_i(t)), u_{j,p+1}(\epsilon_{j,p}(t))).
\end{aligned} \tag{18}$$

Then in terms of Lemma 1, it is obviously obtained that

$$\mathscr{V}_{i,p}(\epsilon_i(t)) \leq -r(\epsilon_i(t), u_{i,p+1}(\epsilon_i(t)), u_{j,p+1}(\epsilon_{j,p}(t))) \leq 0, \tag{19}$$

where $\mathscr{V}_{i,p}(\epsilon_i(t)) = 0$ as $\epsilon_i(t) = 0$. Therefore, the tracking error $\epsilon_i(t)$ will approach 0 when it reaches a sufficiently large value at t. Thus, each node in the graph $\mathfrak{G}$ will achieve group consensus.

Algorithm 1 Data-Based Asynchronous PG-Based Algorithm

Initialize: The iteration index $p = 0$, the iteration period T_i, and the admissible policies $(u_{1,0}(\cdot), u_{2,0}(\cdot), \ldots, u_{n,0}(\cdot))$.

Step 1: Choose nonempty set $M_p = \{i \in \mathbb{N} \mid p \bmod T_i = 0\}$, and $\mathbb{N}\backslash M_p$ represents the set of unactivated agents.

Step 2: Agent i is updated by

For $i \in M_p$ do

Policy Evaluation: The Q-function can be derived by considering the action $u_{i,p}(\cdot)$ taken by agent i and the actions $u_{j,p}$ taken by its neighbors, thus it is shown as

$$\mathscr{Q}_{i,p}(\epsilon_i(t), u) = r(\epsilon_i(t), u_{j,p}(\epsilon_{j,p}(t)), u) + \mathscr{Q}_{i,p}(\epsilon_i(t+1), u_{i,p}(\epsilon_{i,p}(t+1))). \tag{20}$$

Policy Improvement: Improve the control policy based on the action gradient of the above Q-function with following equation:

$$u_{i,p+1}(\epsilon_i(t)) = u_{i,p}(\epsilon_i(t)) - \rho_i \nabla_u \mathscr{Q}_{i,p}(\epsilon_i(t), u_{i,p}(\epsilon_i(t))), \tag{21}$$

in which ρ_i is a constant and named as the learning factor. The symbol ∇_u represents the first-order partial derivative with regard to the direction of u.

For $i \in \mathbb{N}\backslash M_p$ do

Policy Maintain: The agent with rapid processing capabilities, remains in a state of waiting, without making any alterations to the the Q-function and its policy, which can be describe as

$$u_{i,p+1}(\epsilon_i(t)) = u_{i,p}(\epsilon_i(t)), \tag{22}$$

$$\mathscr{Q}_{i,p+1}(\epsilon_i(t), u) = \mathscr{Q}_{i,p}(\epsilon_i(t), u). \tag{23}$$

Step 3: The iteration $p \leftarrow p + 1$, then return to step 1.

3.3. Convergence Analysis

The theorem presented in this part proves the convergence of Algorithm 1, which means that $\mathscr{Q}_{i,p}(\epsilon_i(t), u)$ and policy $u_{i,p}(\epsilon_i(t))$ of agent i can converge to the optimal solution.

Theorem 2. *Under Assumptions 1 and 2, if the initial control laws* $(u_{1,0}(\cdot), u_{2,0}(\cdot), \ldots, u_{n,0}(\cdot))$ *are admissible,* $u_{i,p}(\cdot)$ *and* $\mathscr{Q}_{i,p}(\epsilon_i(t), u)$ *are calculated by Algorithm 1. Then it holds that*

(1) $\lim_{p\to\infty} u_{i,p}(\epsilon_i(t)) = u_i^*(\epsilon_i(t))$.

(2) $\lim_{p\to\infty} \mathscr{Q}_{i,p}(\epsilon_i(t), u) = \mathscr{Q}_i^*(\epsilon_i(t), u)$.

Proof. This theorem is connected to Theorem 1. By utilizing (9) and (10), it is possible to deduce that

$$\begin{aligned} \mathscr{V}_{i,p+1}(\epsilon_i(t)) =& \mathscr{V}_{i,p+1}(\epsilon_i(t+1)) + r(\epsilon_i(t), u_{i,p+1}(\epsilon_i(t)), u_{j,p+1}(\epsilon_{j,p}(t))) \\ =& \mathscr{H}_i(\epsilon_i(t), u_{i,p+1}(\epsilon_i(t)), \mathscr{V}_{i,p}) - \mathscr{V}_{i,p+1}(\epsilon_i(t+1)) + \mathscr{V}_{i,p}(\epsilon_i(t)) + \mathscr{V}_{i,p+1}(\epsilon_i(t+1)) \\ =& \mathscr{H}_i(\epsilon_i(t), u_{i,p+1}(\epsilon_i(t)), \mathscr{V}_{i,p}) + \mathscr{V}_{i,p}(\epsilon_i(t)). \end{aligned} \tag{24}$$

Then according to (20), it can yield that

$$\mathscr{V}_{i,p+1}(\epsilon_i(t)) \leq \mathscr{V}_{i,p}(\epsilon_i(t)). \tag{25}$$

According to (12), it may be inferred that

$$\begin{aligned} \mathscr{Q}_{i,p+1}(\epsilon_i(t), u) =& r(\epsilon_i(t), u_{j,p+1}(\epsilon_j(t), u)) + \mathscr{V}_{i,p+1}(\epsilon_i(t+1)) \\ \leq& r(\epsilon_i(t), u_{j,p}(\epsilon_j(t)), u) + \mathscr{V}_{i,p}(\epsilon_i(t+1)). \end{aligned} \tag{26}$$

From this, it can be obtained that $\mathscr{Q}_{i,p+1}(\epsilon_i, u) \leq \mathscr{Q}_{i,p}(\epsilon_i, u)$. Therefore, the associated Q-function is decreasing for each agent $i \in N$, and as index $p \to \infty$, the Q-function will converge to a minimum value. So it can

be deduced that

$$\begin{aligned}\mathscr{Q}_{i,p}(\epsilon_i(t),u) =& r(\epsilon_i(t),u,u_{j,p}(\epsilon_j(t))) + \mathscr{V}_{i,p}(\epsilon_i(t+1))\\ \geq& r\big(\epsilon_i(t),u,u_j^*(\epsilon_j(t))\big) + \mathscr{V}_i^*(\epsilon_i(t+1))\\ \geq& \mathscr{Q}_i^*(\epsilon_i(t),u).\end{aligned} \tag{27}$$

Therefore, the expression $\lim_{p\to\infty}\mathscr{Q}_{i,p}(\epsilon_i(t),u)=\mathscr{Q}_i^*(\epsilon_i(t),u)$ defines the limit of the Q-function as $p\to\infty$, indicating that the optimal value can be regarded as the convergent point. Furthermore, the partial derivative of $\mathscr{Q}_{i,p}(\cdot)$ obeys the subsequent equation:

$$\lim_{p\to\infty}\nabla_u\mathscr{Q}_{i,p}(\epsilon_i(t),u_{i,p}(\epsilon_i(t)))=0, \tag{28}$$

where $u=u_i^\infty(\epsilon_i(t))$ and $u_i^\infty(\cdot)=\lim_{p\to\infty}u_i^p(\cdot)$. By computing the partial derivative of both sides of Equation (12) with regard to p as $p\to\infty$, it can be deduced that

$$\begin{aligned}\nabla_\pi\mathscr{Q}_i^\infty(\epsilon_i(t),\pi) &= \nabla_\pi(r(\epsilon_i(t),\pi,\pi_j)+\mathscr{V}_i^\infty(\epsilon_i(t+1)))\\ &= \nabla_\pi\big(r\big(\epsilon_i(t),\pi,u_j^*\big)+\mathscr{V}_i^*(\epsilon_i(t+1))\big)\\ &= 0,\end{aligned} \tag{29}$$

in which $\lim_{p\to\infty}u_{j,p}(\epsilon_i(t))=\pi_j=u_j^*$, $\lim_{p\to\infty}u_{i,p}(\epsilon_i(t))=\pi$. Consequently, it is capable of being inferred that $\mathscr{Q}_{i,\infty}(\epsilon_i,u)=\mathscr{Q}_i^*(\epsilon_i,u)$. This suggests that 2) in Theorem 2 holds. Meanwhile, as stated in condition 1) of Theorem 2, it can be demonstrated through the use of a proof by contradiction, which assumes that $\lim_{p\to\infty}u_{i,p}(\epsilon_i(t))\neq u_i^*(\epsilon_i(t))$, then

$$\begin{aligned}u_{i,\infty+1}(\epsilon_i(t)) &= u_{i,\infty}(\epsilon_i(t))-\rho_i\nabla_u\mathscr{Q}_{i,\infty}(\epsilon_i(t),u_{i,\infty}(\epsilon_i(t)))\\ &\neq u_{i,\infty}(\epsilon_i(t)),\end{aligned} \tag{30}$$

which is contradicted with (29). Accordingly, we have $\lim_{p\to\infty}u_{i,p}(\epsilon_i(t))=u_i^*(\epsilon_i(t))$. □

Consider the global Lyapunov candidate function composed of the state value functions of all agents $L_p(t)=\sum_{i=1}^N\mathscr{V}_{i,p}(\epsilon_i(t))$. In each iteration p of Algorithm 1, agents are divided into the update set M_p and the maintenance set $\mathbb{N}\setminus M_p$. For $i\in M_p$ (updating agents), according to the proofs of Lemma 1, Theorems 1 and 2, the policy improvement ensures that their local Lyapunov difference satisfies $\Delta\mathscr{V}_{i,p}(_i(t))\leq -r(\cdot)\leq 0$. For $i\in\mathbb{N}\setminus M_p$ (policy-maintaining agents), their control policies and Q-functions remain unchanged, i.e., $u_{i,p+1}=u_{i,p}$, $\mathscr{Q}_{i,p+1}(\epsilon_i(t),u)=\mathscr{Q}_{i,p}(\epsilon_i(t),u)$. Their local state value functions remain unchanged in this iteration: $\mathscr{V}_{i,p+1}(_i(t))=\mathscr{V}_{i,p}(_i(t))$, so their Lyapunov difference is zero, i.e., $\Delta\mathscr{V}_{i,p}(_i(t))=0$. Therefore, the difference of the global Lyapunov function is

$$\Delta L_p(t)=\sum\nolimits_{i\in M_p}\Delta\mathscr{V}_{i,p}(\epsilon_i(t))+\sum\nolimits_{i\in\mathbb{N}\setminus M_p}\Delta\mathscr{V}_{i,p}(\epsilon_i(t))\leq\sum\nolimits_{i\in M_p}(-r(\cdot))+0\leq 0$$

This indicates that regardless of the states of the agents in the "policy maintenance" stage, the global Lyapunov function $L_p(t)$ always shows a non-increasing trend. The asynchronous update ensures that within the infinite iteration, each agent will be accessed infinitely many times due to its finite update period T_i. This means that no agent can remain in the "policy maintenance" stage indefinitely. Therefore, on the global time scale, the system energy represented by $L_p(t)$ will continue to dissipate until convergence, thus ensuring that the tracking error $\epsilon_i(t)$ can achieve asymptotic stability and group consensus based on the asynchronous learning process.

Remark 5. *The work [27] mainly focuses on data-driven OCC for homogeneous MASs, adopting an RL approach based on value functions. Although [21] employs the PG method and proposes an asynchronous version, its system model still relies on the homogeneous MASs and does not involve the mixed-order dynamics and group consensus control. We propose a designed asynchronous PG algorithm for HeMASs with mixed-order dynamics. Compared with [21,27], the proposed asynchronous mechanism not only considers differences in computational capabilities but also addresses the time mismatch in policy updates and virtual velocity estimation. Moreover, introducing group competition-cooperation mechanism requires agents to handle more complex neighboring information during asynchronous updates, which imposes higher demands on the algorithm convergence.*

4. Implementation of AC Structures

The algorithm that has been suggested is put into effect in this part by employing the AC structures. The structure consists of the actor and the critic. The actor network is utilized to estimate the control policy, and the critic network is responsible for learning the performance index function.

Three-layer neural networks are defined by using neural networks to create actors and critics, which can be shown that

$$\mathcal{F}(e, W) = W^T \sigma(e), \tag{31}$$

where e stands for the input of neural networks, the weighting matrix is defined as $W = \{W_1, W_2, \ldots, W_L\}^T$, and it represents the connections within the neural output layer and hidden layer, then $\sigma(\cdot) = \{\sigma_1(\cdot), \sigma_2(\cdot), \ldots, \sigma_L(\cdot)\}$ denotes the activation function.

To better illustrate the principle of this algorithm, the structure of the designed algorithm and its specific implementation process are shown in Figure 1.

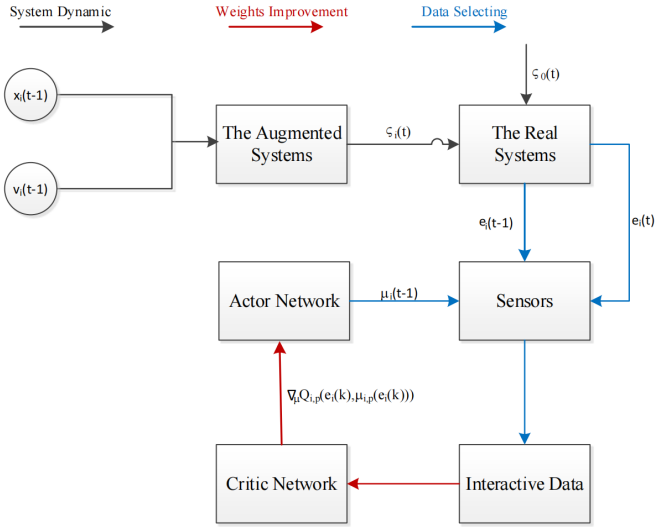

Figure 1. The proposed algorithm's overall process framework.

The algorithm's structure is illustrated in Figure 1, which employs the actual system to calculate the local tracking errors $\epsilon_i(t)$ by utilizing the local state $\varsigma_i(t)$. Additionally, the data set that generated by the system will serve as an experiential collection for the learning process. Furthermore, the control policy will be improved by integrating the real-time data of the sensors as shown in Figure 1. Additionally, there is a critic network that guides the actor's actions by assessing the quality of the control policy, and becomes accustomed to evaluating the Q-function. There exists another actor network that continuously improves the control policy by incorporating feedback from the critic as well.

Remark 6. *The technique of experience replay and an offline data set are employed in the proposed Algorithm 1, which are not discussed in references [35,38]. The data set is created by the system's real-time interaction process.*

4.1. Design of the Critic Networks

The local Q-function is approximated by the critic networks, given by

$$\hat{Q}(\epsilon_i, u) = W_{ci}^{\top} \delta_{ci}(h_{ci}(t)), \tag{32}$$

where $W_{ci}^{\top}$ is the weight matrix, $\delta_{c_i}(\cdot)$ is the activation function, and $h_{ci}(t)$ denotes the input vector. That is $h_{ci}(t) = \left\{\epsilon_i(t), u, u_{N(i)}\left(\epsilon_{N(i)}(t)\right)\right\}$. Q' indicates the target value in the following formulation. The critic network's approximate error performance index is denoted by

$$E_{ci}(t) = \frac{1}{2}\epsilon_i^{\top}(t)\epsilon_i(t), \tag{33}$$

and we need to minimize $E_{ci}(t)$ in (33). Meanwhile, $\epsilon_i(t)$ can be described as

$$\epsilon_i(t) = \hat{Q}(\epsilon_i, u) - Q(\epsilon_i, u) = \hat{Q}(\epsilon_i, u) - r(\epsilon_i, u_j(t), u_i(t)) - \hat{Q}'(\epsilon_i(t+1), \hat{u}_i(t+1)). \tag{34}$$

Therefore, the weight matrix of critic neural networks can be updated with

$$\begin{aligned} W_{ci}^{\top}(p+1) &= W_{ci}^{\top}(p) - \beta_c \frac{\partial E_{ci}(t)}{\partial W_{ci}(t)} \\ &= W_{ci}^{\top}(p) - \beta_c \frac{\partial E_{ci}(t)}{\partial \epsilon_i(t)} \frac{\partial \epsilon_i(t)}{\partial \hat{Q}_i(t)} \frac{\partial \hat{Q}_i(t)}{dW_{ci}(t)} \\ &= W_{ci}^{\top}(p) - \beta_c \delta_{ci}(h_{ci}(t)) \epsilon_i^{\top}(t), \end{aligned} \tag{35}$$

where $\beta_c > 0$ denotes the learning rate.

4.2. Design of the Actor Networks

Then, the control policy is approximated by applying actor networks, which are defined as

$$\hat{u}_i(t) = W_{ai}^{\top} \delta_{ai}(h_{ai}(t)). \tag{36}$$

Just similar to the critic network weight update above, the weight matrices of actor neural networks can be updated by

$$\begin{aligned} W_{ai}^{\top}(p+1) &= W_{ai}^{\top}(p) - \beta_a \frac{\partial \hat{Q}_i(t)}{\partial W_{ai}^{\top}(t)} = W_{ai}^{\top}(p) - \beta_a \frac{\partial \hat{Q}_i(t)}{\partial h_{ci}(t)} \frac{\partial h_{ci}(t)}{\partial \hat{u}_i(t)} \frac{\partial \hat{u}_i(t)}{\partial W_{ai}(t)} \\ &= W_{ai}^{\top}(p) - \beta_a W_{ci}^{\top} \delta'_{ci}(h_{ci}(t)) \frac{\partial h_{ci}(t)}{\partial \hat{u}_i(t)} \delta_{ai}(h_{ai}(t)), \end{aligned} \tag{37}$$

where $\beta_a > 0$ is the learning rate.

4.3. Stability Analysis of AC Networks

Base on (32) and (36), the performance index function and the control policy can be reconstructed as

$$Q(\epsilon_i, \mu) = W_{ci}^{*^{\top}} \delta_{ci}(h_{ci}(t)) + \tau_c, \tag{38}$$

$$u_i(t) = W_{ai}^{*^{\top}} \delta_{ai}(h_{ai}(t)) + \tau_a, \tag{39}$$

in which τ_c and τ_a stand for the reconstruction errors of critic-network and actor-network.

Theorem 3. *Assuming that the optimal solution for neural network weights W_{ci}^* and W_{ai}^* exist, and the reconstruction error τ_c is norm-bounded. That is, $\|W_{ci}^*\| \leq W_{cM}, \|W_{ai}^*\| \leq W_{aM}, \tau_c \leq \tau_{cM}$, and $\|\delta_c^p\| \leq \delta_{cM}, \|\delta_a^p\| \leq \delta_{aM}$, where W_{cM}, W_{aM}, τ_{cM}, δ_{cM} and δ_{aM} are the corresponding upper bounds. Then, the weight estimation errors $\tilde{W}_c(p) = W_c(p) - W_c^*$ and $\tilde{W}_a(p) = W_a(p) - W_a^*$ existing in the AC networks are uniformly ultimately bounded (UUB), which implies that the weight estimation process eventually stabilize.*

Proof. According to (35) and (37), it can be obtained that

$$\tilde{W}_c(p+1) = \tilde{W}_c(p) - \beta_c \delta_c^p \epsilon_c^{pT}(t), \tag{40}$$

$$\tilde{W}_a(p+1) = \tilde{W}_a(p) - \beta_a W_c^{\top}(p) \dot{\delta}_c^p \delta_a^p \gamma, \tag{41}$$

where $\gamma = \frac{\partial h_{c_i}(t)}{\partial \hat{u}_i(t)}$ and $\gamma \leq \gamma_M$. $\dot{\delta}_c^p$ is the derivative of δ_c^p, and $\left\|\dot{\delta}_c^p\right\| \leq \dot{\delta}_{cM}, \left\|\dot{\delta}_a^p\right\| \leq \delta_{aM}$, then we can select the Lyapunov function as

$$L = \frac{L_1}{\beta_c} + \frac{L_2}{\beta_a} = \frac{1}{\beta_c} \operatorname{tr}\left\{\tilde{W}_c^{\top}(p) \tilde{W}_c(p)\right\} + \frac{1}{\beta_a} \operatorname{tr}\left\{\tilde{W}_a^{\top}(p) \tilde{W}_a(p)\right\}, \tag{42}$$

and the difference of L_1 can be conducted that

$$\Delta L_1 = \operatorname{tr}\left\{\tilde{W}_c^{\top}(p+1) \tilde{W}_c(p+1)\right\} - \operatorname{tr}\left\{\tilde{W}_c^{\top}(p) \tilde{W}_c(p)\right\}. \tag{43}$$

Based on (32) and (38),

$$\epsilon_c^p = W_c^{\top}(p) \delta_c^p - W_c^{*^{\top}} \delta_c^p - \tau_c. \tag{44}$$

Let $\psi_c^p = W_c^\top(p)\delta_c^p - W_c^{*\top}\delta_c^p = \tilde{W}_c^\top(p)\delta_c^p$, then we can get with (40) and (44) that

$$\tilde{W}_c(p+1) = \tilde{W}_c(p) - \beta_c\delta_c^p\left(\psi_c^{p\top} - \tau_c^\top\right). \tag{45}$$

Therefore, according to the Cauchy-Schwarz inequality, ΔL_1 can be deduced that

$$\begin{aligned}
\Delta L_1 =& \operatorname{tr}\Big\{\tilde{W}_c^\top(p)\tilde{W}_c(p) + \beta_c^2(\delta_c^p(\psi_c^p - \tau_c))^\top \times \delta_c^p\left(\psi_c^{p\top} - \tau_c^\top\right) \\
&- 2\beta_c\tilde{W}_c^\top(p)\delta_c^p\left(\psi_c^{p\top} - \tau_c^\top\right)\Big\} - \operatorname{tr}\Big\{\tilde{W}_c^\top(p)\tilde{W}_c(p)\Big\} \\
\le& \operatorname{tr}\Big\{\beta_c\Big(\beta_c\left(\delta_c^p\left(\psi_c^{p\top} - \tau_c^\top\right)\right)^\top\delta_c^p\left(\psi_c^{p\top} - \tau_c^\top\right) - 2\psi_c^p\left(\psi_c^{p\top} - \tau_c^\top\right)\Big)\Big\} - \operatorname{tr}\{\tilde{w}_c^\top(p)\tilde{w}_c(p)\} \\
\le& \beta_c\Big(-2\|\psi_c^p\|^2 + 2\psi_c^p\tau_c^\top + 2\beta_c\Big(\left\|\delta_c^p\psi_c^{p\top}\right\|^2\Big) + \left\|\delta_c^p\tau_c^\top\right\|^2\Big) \\
\le& \beta_c\Big(-2\|\psi_c^p\|^2 + 2\psi_c^p\tau_c^\top + 2\beta_c\Big(\|\delta_c^p\|^2\left\|\psi_c^{p\top}\right\|^2\Big)\Big) + \beta_c\|\delta_c^p\|^2\|\tau_c\|^2 \\
=& \beta_c\Big(-\left(1 - 2\beta_c\|\delta_c^p\|^2\right)\|\psi_c^p\|^2 + \left(1 + 2\beta_c\|\delta_c^p\|^2\right)\|\tau_c^p\|^2\Big),
\end{aligned} \tag{46}$$

by the same analysis, it can be concluded that

$$\Delta L_2 \le \beta_a\left(-\left(\|\gamma\|^2 - \beta_a\|\gamma\|^2\|\delta_a^p\|^2\right)\left\|W_c^\top(p)\dot{\delta}_c^p\right\|^2 + \frac{1}{2}\|\psi_a^p\|^4 + \|\gamma\|^4 + \left\|w_c^\top(p)\dot{\delta}_c^p\right\|^2 + \frac{1}{2}\|w_c^\top(p)\dot{\delta}_c^p\|^4\right). \tag{47}$$

Let

$$\omega_M \triangleq \frac{1}{2}\left(W_{cM}^\top\dot{\delta}_{cM}\right)^4, \tag{48}$$

and $\|\psi_c^p\| \le \sqrt{\frac{\omega_M}{1-2\beta_c\psi_{cM}^2}}$ with $\beta_c \in \left[0, \frac{1}{2\delta_{cM}^2}\right]$, $\beta_a \in \left[0, \frac{1}{\delta_{aM}^2}\right]$, and then, ΔL yields

$$\begin{aligned}
\Delta L =& \frac{1}{\beta_c}\Delta L_1 + \frac{1}{\beta_a}\Delta L_2 \\
\le& -\left(1 - 2\beta_c\|\delta_c^p\|^2\right)\|\psi_c^p\|^2 + \left(1 + 2\beta_c\|\delta_c^p\|^2\right)\|\tau_c^p\|^2 - \left(\|\gamma\|^2 - \beta_a\|\gamma\|^2\|\delta_a^p\|^2\right)\left\|W_c^\top(p)\dot{\delta}_c^p\right\|^2 \\
&+ \frac{1}{2}\|\psi_a^p\|^4 + \|\gamma\|^4 + \left\|w_c^\top(p)\dot{\delta}_c^p\right\|^2 + \frac{1}{2}\|w_c^\top(p)\dot{\delta}_c^p\|^4 - \left(1 - 2\beta_c\|\delta_c^p\|^2\right)\|\psi_c^p\|^2 \\
&- \left(\|\gamma\|^2 - \beta_a\|\gamma\|^2\|\delta_a^p\|^2\right)\left\|w_c^\top(p)\dot{\delta}_c^p\right\|^2 + \left(1 + 2\beta_c\|\delta_c^p\|^2\right)\|\tau_c^p\|^2 + \frac{1}{2}\|\psi_a^p\|^4 + \|\gamma\|^4 + \left\|W_c^\top(p)\dot{\delta}_c^p\right\|^2 \\
&+ \frac{1}{2}\left\|W_c^\top(p)\dot{\delta}_c^p\right\|^4 + \frac{1}{2}\|\psi_a^p\|^4 + \|\gamma\|^4 + \left\|w_c^\top(p)\dot{\delta}_c^p\right\|^2 + \frac{1}{2}\|w_c^\top(p)\dot{\delta}_c^p\|^4 - \left(1 - 2\beta_c\|\delta_c^p\|^2\right)\|\psi_c^p\|^2 \\
&- \left(\|\gamma\|^2 - \beta_a\|\gamma\|^2\|\delta_a^p\|^2\right)\left\|w_c^\top(p)\dot{\delta}_c^p\right\|^2 + \left(1 + 2\beta_c\|\delta_c^p\|^2\right)\|\tau_c^p\|^2 + \frac{1}{2}\|\psi_a^p\|^4 + \|\gamma\|^4 + \left\|W_c^\top(p)\dot{\delta}_c^p\right\|^2 \\
&+ \frac{1}{2}\left\|W_c^\top(p)\dot{\delta}_c^p\right\|^4 \\
\le& \left(2\beta_c\|\delta_c^p\|^2 - 1\right)(1 - 2\beta_c\psi_{cM}^2)^{-1}\omega_M + \left(1 + 2\beta_c\|\delta_c^p\|^2\right)\|\tau_c^p\|^2 - \left(\|\gamma\|^2 - \beta_a\|\gamma\|^2\|\delta_a^p\|^2\right)\left\|W_c^\top(p)\dot{\delta}_c^p\right\|^2 \\
&+ \frac{1}{2}\|\psi_a^p\|^4 + \|\gamma\|^4 + \left\|w_c^\top(p)\dot{\delta}_c^p\right\|^2 + \frac{1}{2}\|w_c^\top(p)\dot{\delta}_c^p\|^4 - \left(1 - 2\beta_c\|\delta_c^p\|^2\right)(1 - 2\beta_c\psi_{cM}^2)^{-1}\omega_M \\
&- \left(\|\gamma\|^2 - \beta_a\|\gamma\|^2\|\delta_a^p\|^2\right)\left\|w_c^\top(p)\dot{\delta}_c^p\right\|^2 + \left(1 + 2\beta_c\|\delta_c^p\|^2\right)\|\tau_c^p\|^2 + \frac{1}{2}\|\psi_a^p\|^4 + \|\gamma\|^4 + \left\|W_c^\top(p)\dot{\delta}_c^p\right\|^2 \\
&+ \frac{1}{2}\|\psi_a^p\|^4 + \|\gamma\|^4 + \left\|w_c^\top(p)\dot{\delta}_c^p\right\|^2 + \frac{1}{2}\|w_c^\top(p)\dot{\delta}_c^p\|^4 - \left(1 - 2\beta_c\|\delta_c^p\|^2\right)(1 - 2\beta_c\psi_{cM}^2)^{-1}\omega_M \\
&- \left(\|\gamma\|^2 - \beta_a\|\gamma\|^2\|\delta_a^p\|^2\right)\left\|w_c^\top(p)\dot{\delta}_c^p\right\|^2 + \left(1 + 2\beta_c\delta_{cM}^2\right)\tau_{cM}^2 + \frac{1}{2}\psi_{aM}^4 + \gamma_M^4 + \left(W_{cM}^\top\dot{\delta}_{cM}\right)^2 \\
&+ \frac{1}{2}\left\|W_c^\top(p)\dot{\delta}_c^p\right\|^4 + \frac{1}{2}\left(W_{cM}^\top\dot{\delta}_{cM}\right)^4.
\end{aligned}$$

Combining like terms, we obtain $\Delta L < 0$. Hence, the weight estimation errors $\tilde{W}_c(p)$ and $\tilde{W}_a(p)$ are UUB. □

The stability bounds of the control system are essentially related to the approximation accuracy of the actor-critic neural networks. As elaborated in the Lyapunov analysis (e.g., in the derivation of the ultimate bound ω_M

in (48)), the reconstruction errors τ_c and τ_a of the critic and actor networks, respectively, manifest as persistent, bounded disturbances in the weight update dynamics. According to the theory of perturbed systems and the Lyapunov method, such bounded disturbances prevent the system from achieving asymptotic stability but ensure UUB, where the size of the ultimate bound is a continuous function of the disturbance amplitude.

This relationship is a well-established principle in the adaptive dynamic programming (ADP). Reference [26] emphasizes that the convergence of ADP algorithms is within a neighborhood of the optimal solution, the size of which is determined by the approximation error of the critic network. Similarly, reference [37] analytically demonstrates that the residual error from the neural network approximation directly influences the closed-loop stability margin. Therefore, in order to obtain stricter stability bounds and better steady-state performance, it is necessary to use neural networks with sufficient representational capacity to minimize the inherent reconstruction errors τ_c and τ_a as much as possible.

Remark 7. *In the concurrent update process of the actor and critic networks, the critic network provides an evolving estimate of the Q-function, which guides the actor's policy improvement. Although the critic's weights are updated asynchronously and may temporarily be inaccurate, the Lyapunov-based analysis in Theorem 3 ensures that the weight estimation errors of both networks are UUB. This implies that the critic's evaluation error remains within a bounded region, thereby preventing the actor from being misled by large estimation errors. Moreover, the use of experience replay and offline datasets helps to stabilize the learning by reducing the correlation in the data and providing more consistent gradient estimates. Therefore, even under asynchronous and concurrent updates, the actor's policy learning remains stable and converges to a near-optimal solution.*

5. Simulation Results

The dynamic of HeMASs with the communication topology graph (Figure 2) is represented by the following:

$$\begin{cases} \begin{cases} x_i(t+1) = \mathfrak{A}x_i(t) + \mathfrak{B}v_i(t) \\ v_i(t+1) = \mathfrak{C}v_i(t) + \phi_i u_i(t) \end{cases}, & i \in \beta_1 \\ x_i(t+1) = \mathfrak{A}x_i(t) + \widehat{\phi}_i u_i(t), & i \in \beta_2 \end{cases}, \tag{49}$$

and the leader dynamics is denoted as

$$\begin{cases} x_0(t+1) = \mathscr{H}x_0(t) + \mathfrak{B}v_0(t) \\ v_0(t+1) = \mathfrak{C}v_0(t) \end{cases}. \tag{50}$$

Then the transformed homogeneous augmented systems can be derived as

$$\begin{cases} \begin{cases} \varsigma_i(t+1) = \Xi\varsigma_i(t) + \xi_i u_i(t), & i \in \beta_1 \\ \varsigma_i(t+1) = \Xi\varsigma_i(t) + \widetilde{\xi}_i \widetilde{u}_i(t), & i \in \beta_2 \end{cases} \\ \varsigma_0(t+1) = \Xi\varsigma_0(t), \end{cases}. \tag{51}$$

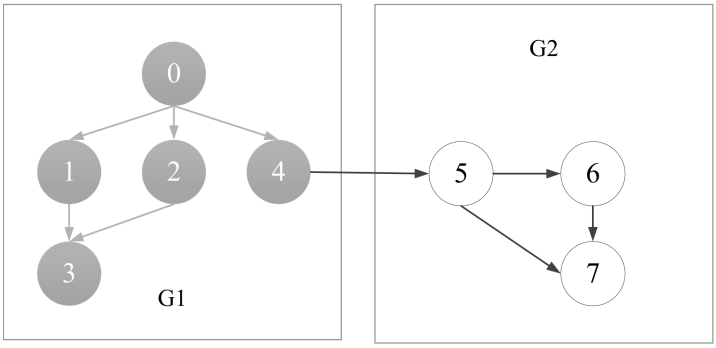

Figure 2. Communication topology of the MASs: The agents are divided into two groups, in which the one in G1 represents the second-order agent, and one in G2 represents the first-order agent.

Assume that the system matrices $\Xi = \begin{pmatrix} 1.967 & 0.798 \\ 0 & 0.9896 \end{pmatrix}$, and $\xi_1 = (0, 1.11)^\top, \xi_2 = (0, 0.82)^\top, \xi_3 = (0, 0.91)^\top, \xi_4 = (0, 0.75)^\top$, the symbol $\mathrm{diag}\{\}$ in the following represents a diagonal matrix, and all elements outside the main diagonal are zeros, so the second-order input matrices is that $\widetilde{\xi}_5 = \mathrm{diag}\{0.779, 0.779\}$, $\widetilde{\xi}_6 =$

$\mathrm{diag}\{0.51, 0.51\}$, $\widetilde{\xi}_7 = \mathrm{diag}\{0.86, 0.86\}$. If agent i has the ability to directly receive information from the leader 0 directly, we set $b_i = 1$, else $b_i = 0$. So we set $b_1 = b_2 = b_4 = 1$, and the neighboring elements among the agents are selected as $a_{31} = a_{32} = a_{54} = a_{65} = a_{75} = a_{76} = 1$. In addition, the cooperative-competitive strength $S = \begin{pmatrix} S_{\mathrm{ss}} & S_{\mathrm{sf}} \\ S_{\mathrm{fs}} & S_{\mathrm{ff}} \end{pmatrix}$ is chosen as

$$S_{\mathrm{ss}} = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}, \quad S_{sf} = 0_{4\times 3}, S_{fs} = \begin{pmatrix} 0 & 0 & 0 & -1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}, \quad S_{ff} = \begin{pmatrix} 0 & 0 & 0 \\ 1 & 0 & 0 \\ 1 & 1 & 0 \end{pmatrix}.$$

This yields the principle of cooperation within one group and competition between different groups. In addition, we set the learning rates as $\beta_c = \beta_a = 0.003$. Choose the asynchronous update period $T1 = T2 = T3 = T4 = 1$, $T5 = T6 = T7 = 3$. The initial velocity of the leader is set at $v_0 = 0.63$, as shown in Figure 3. All followers in G1 reach the same velocity state as the leader. According to the information provided in Figure 4, it is evident that the positions of the agents in G1 are consistent with those of the leader, but the agents in G2 converge to a different state from that of G1 on their own. Under the asynchronous RL algorithm, the convergence rates of the local tracking errors of all followers are shown in Figure 5. The convergence rates of the first-order agents (nodes 5, 6, 7) are significantly faster than that of the second-order agents (nodes 1, 2, 3, 4), which indicates that Algorithm 1 successfully achieves optimal group consensus in the two state dimensions.

Figures 6 and 7 illustrate the convergence of the weights of the critic network and the actor network. From Figures 8 and 9, it can be seen that the convergence of control policies among the mixed-order agents is ensured. After approximately 200 iterations, the weights and control inputs are stabilized, indicating that the policy networks have converged.

The weights of the first-order agent and the fluctuation amplitude of the control input were stabilized after 60 iterations, which was slightly smaller than that of the second-order agent. It was also stabilized after 80 iterations, indicating that the second-order system brought additional complexity. The fluctuation amplitude of the weights and control inputs for the first-order agents is stabilized after 60 iterations, which is slightly smaller than that of the second-order agents stabilized after 80 iterations. This means that the second-order system introduces additional complexity.

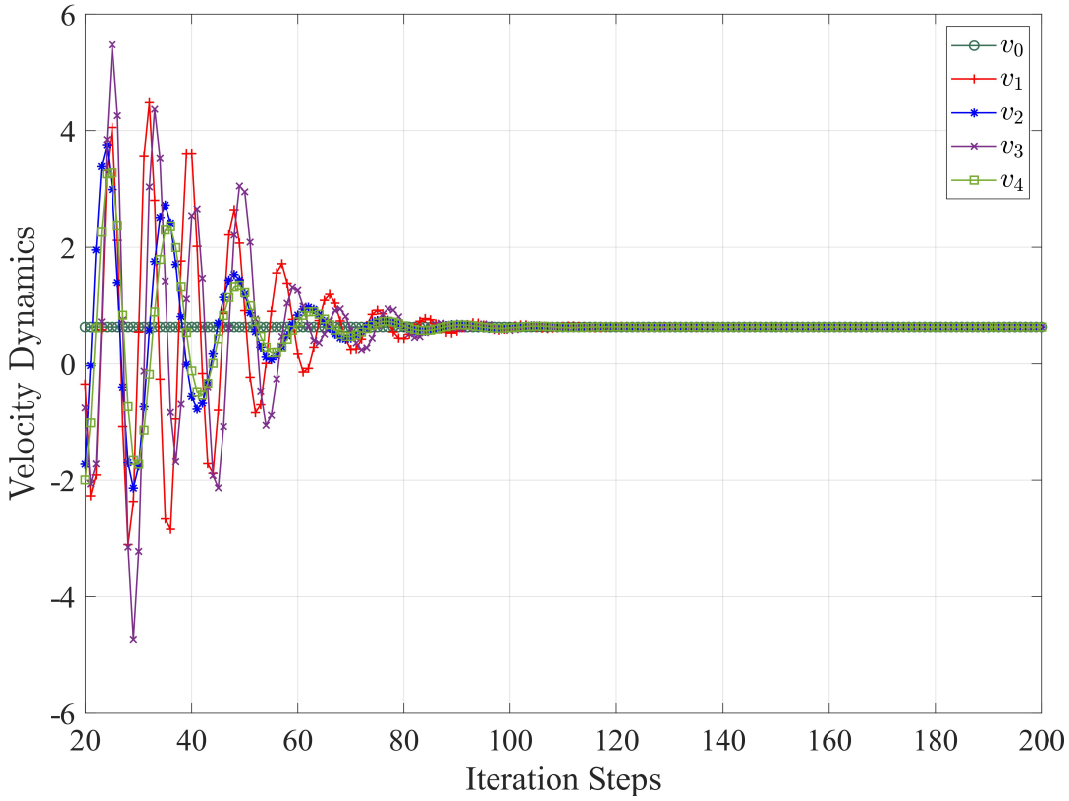

Figure 3. Group consensus of velocity states for second-order agents.

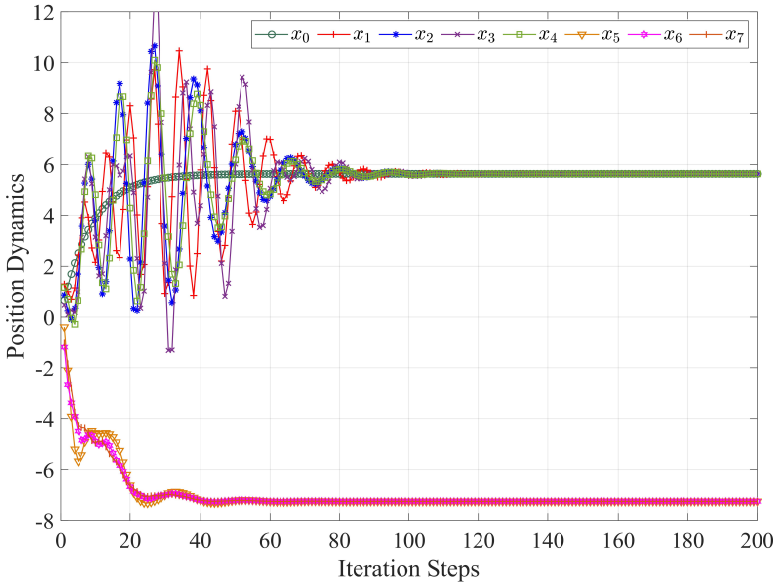

Figure 4. Group consensus of position states for all agents.

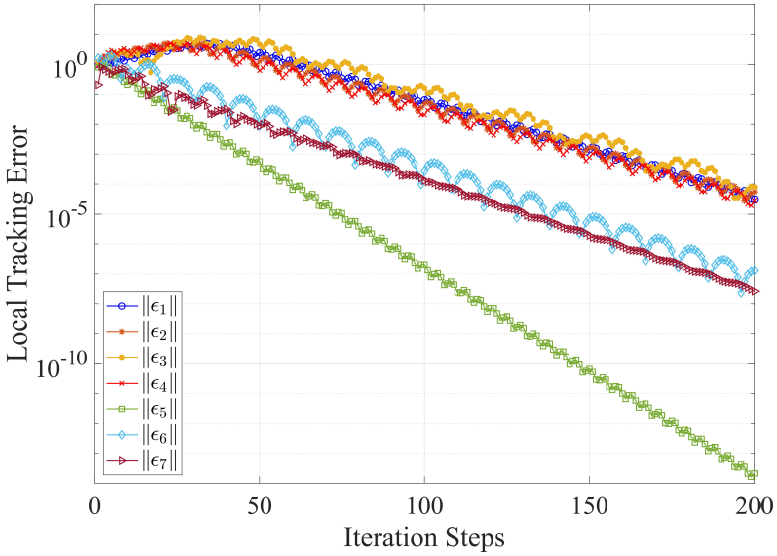

Figure 5. The local tracking errors in the systems (4) by the asynchronous Algorithm 1.

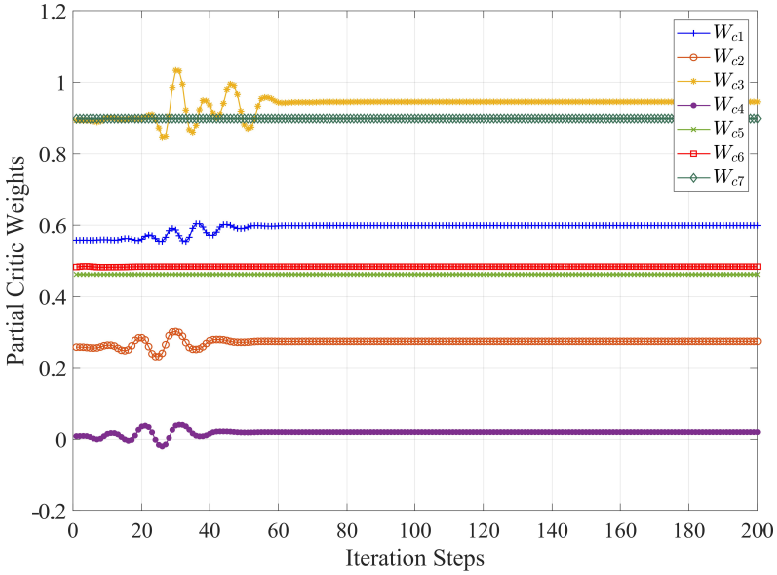

Figure 6. Convergence of the critic weights for all agents.

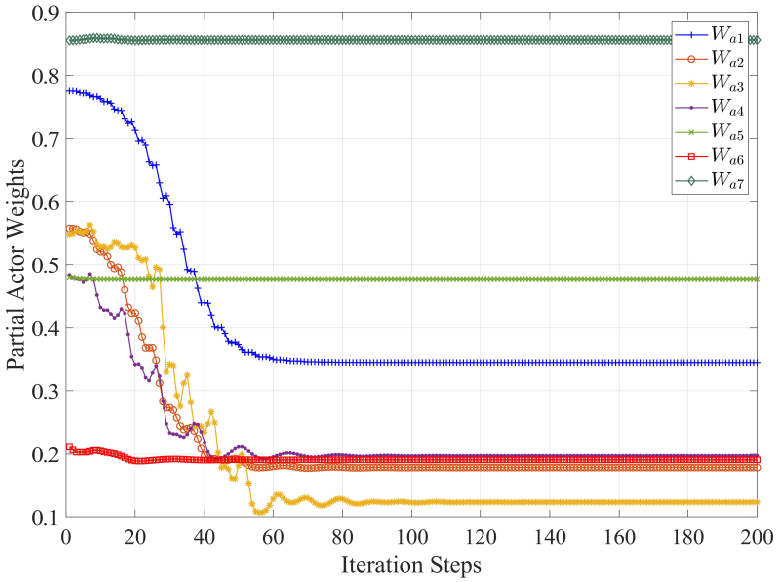

Figure 7. Convergence of the actor weights for all agents.

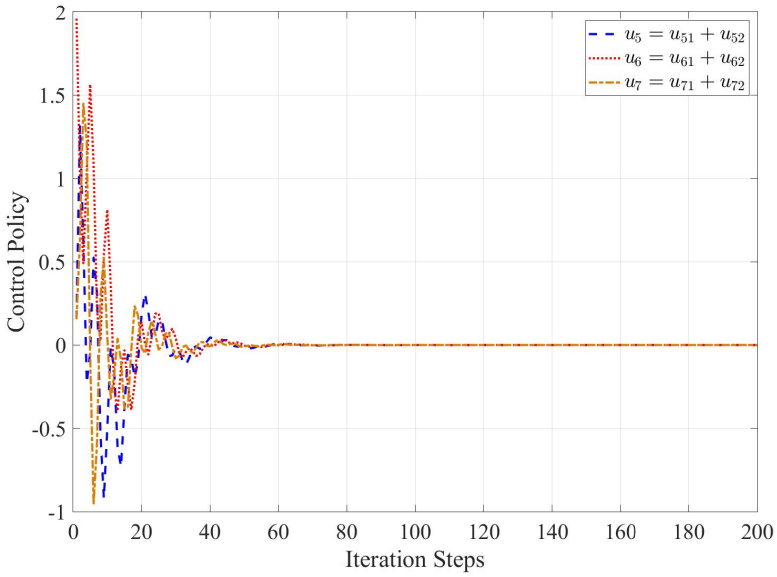

Figure 8. The control policies of the first-order agents.

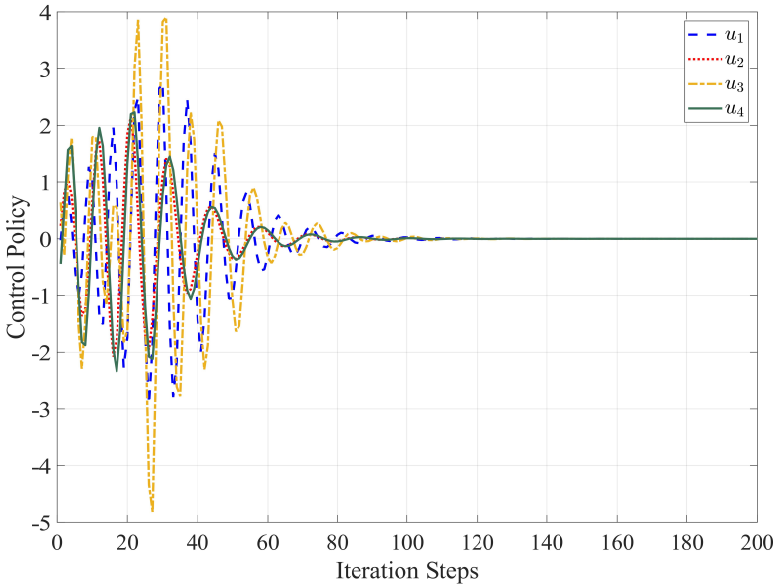

Figure 9. The control policies of the second-order agents.

Let the global tracking error $\epsilon = col(\epsilon_i)$. To quantitatively evaluate the efficiency of the proposed asynchronous algorithm, we compare its convergence performance with the synchronous algorithm in [39]. The convergence threshold is set to $|\epsilon| \leq 10^{-3}$. As shown in Figure 10, the proposed method achieves the expected accuracy in only 20 iterations, which is much faster than the 160 iterations required by the benchmark synchronous method [39].

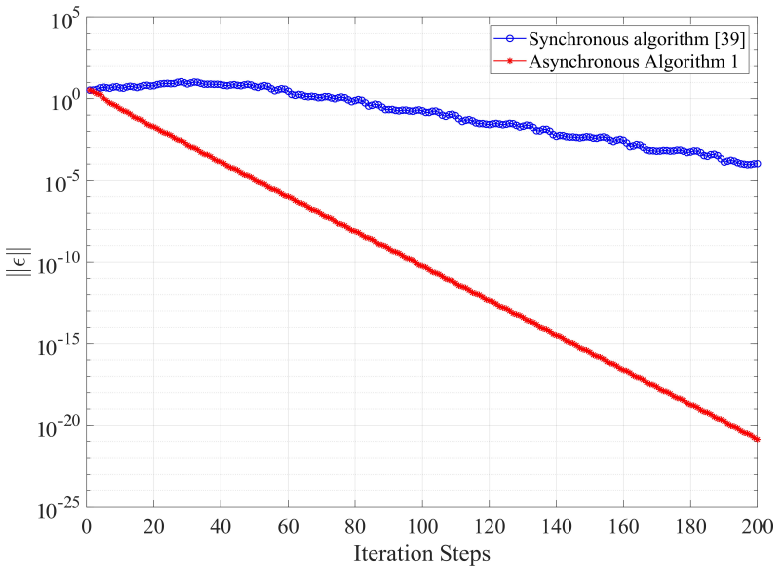

Figure 10. Global tracking error's Euler norms $\|\epsilon\|$.

This means that the number of iterations required to reach convergence has decreased by 87.5%. This result strongly indicates that the asynchronous strategy can significantly improve the learning efficiency and convergence rate of HeMASs by effectively adapting to the differences in the computing capabilities of each agent.

Remark 8. *The proposed algorithm is different from the periodic policy update method in [39]. Due to the differences in the computing capabilities of agents, asynchronous updates inevitably arise in practice, which can affect the performance of the entire system. The asynchronous PG algorithm can effectively solve this problem.*

6. Conclusions

The aim of this study is to solve the OCGCC problem for discrete-time HeMASs with unknown dynamics using a PG method. The proposed data-based asynchronous PG algorithm is executed via the AC scheme, and its stability is analyzed by the Lyapunov approach. It uses a mixed-order heterogeneous framework and introduces cooperation-competition strength factors for intra-group cooperation and inter-group competition. The asynchronous update method addresses the issue of agents' computational capability discrepancies, and the experience replay strategy enhances data utilization. Future research should incorporate more comprehensive cooperative and competitive dynamics into the system.

Author Contributions

L.J.: Conceptualization, Methodology, Investigation, Formal Analysis, Writing Original Draft; X.P.: Data Curation, Formal Analysis, Writing Original Draft; J.L.: Visualization, Writing, Review and Editing. All authors have read and agreed to the published version of the manuscript.

Funding

This work is supported by the National Natural Science Foundation of China under Grant 62276036; in part by the Innovation and Development Joint Fund Project of Chongqing Natural Science Foundation under Grant No. CSTB2024NSCQ-LZX0118.

Institutional Review Board Statement

Not applicable.

Informed Consent Statement

Not applicable.

Data Availability Statement

Not applicable.

Conflicts of Interest

The authors declare no conflict of interest.

Use of AI and AI-Assisted Technologies

No AI tools were utilized for this paper

References

1. Lin, Z.; Wang, L.; Han, Z.; et al. Distributed Formation Control of Multi-Agent Systems Using Complex Laplacian. *IEEE Trans. Autom. Control* **2014**, *59*, 1765–1777.
2. Prodanovic, M.; Green, T. High-Quality Power Generation Through Distributed Control of a Power Park Microgrid. *IEEE Trans. Ind. Electron.* **2006**, *53*, 1471–1482.
3. Gao, W.; Jiang, Z.P.; Ozbay, K. Data-Driven Adaptive Optimal Control of Connected Vehicles. *IEEE Trans. Intell. Transp. Syst.* **2017**, *18*, 1122–1133.
4. Olfati-Saber, R.; Murray, R. Consensus Problems in Networks of Agents With Switching Topology and Time-Delays. *IEEE Trans. Autom. Control* **2004**, *49*, 1520–1533.
5. Lesser, V.; Tambe, M.; Ortiz, C.L. *Distributed Sensor Networks: A Multiagent Perspective*; Kluwer Academic Publishers: Dordrecht, The Netherlands, 2003.
6. Yang, R.; Zhang, H.; Feng, G.; et al. Robust Cooperative Output Regulation of Multi-Agent Systems via Adaptive Event-Triggered Control. *Automatica* **2019**, *102*, 129–136.
7. Qin, J.; Yu, C.; Gao, H. Coordination for Linear Multiagent Systems With Dynamic Interaction Topology in the Leader-Following Framework. *IEEE Trans. Ind. Electron.* **2014**, *61*, 2412–2422.
8. Wang, N.; Gao, Y.; Zhao, H.; et al. Reinforcement Learning-Based Optimal Tracking Control of an Unknown Unmanned Surface Vehicle. *IEEE Trans. Neural Netw. Learn. Syst.* **2021**, *32*, 3034–3045.
9. Shi, H.; Shi, L.; Xu, M.; et al. End-to-End Navigation Strategy With Deep Reinforcement Learning for Mobile Robots. *IEEE Trans. Ind. Inform.* **2020**, *16*, 2393–2402.
10. Li, L.; Wu, D.; Huang, Y.; et al. A Path Planning Strategy Unified With a COLREGS Collision Avoidance Function Based on Deep Reinforcement Learning and Artificial Potential Field. *Appl. Ocean. Res.* **2021**, *113*, 102759.
11. Zhou, W.; Liu, Z.; Li, J.; et al. Multi-Target Tracking for Unmanned Aerial Vehicle Swarms Using Deep Reinforcement Learning. *Neurocomputing* **2021**, *466*, 285–297.
12. Jiang, Y.; Jiang, Z.P. Computational Adaptive Optimal Control for Continuous-Time Linear Systems With Completely Unknown Dynamics—ScienceDirect. *Automatica* **2012**, *48*, 2699–2704.
13. Zhang, J.; Zhang, H.; Feng, T. Distributed Optimal Consensus Control for Nonlinear Multiagent System With Unknown Dynamic. *IEEE Trans. Neural Netw. Learn. Syst.* **2018**, *29*, 3339–3348.
14. Xiong, H.; Chen, G.; Ren, H.; et al. Broad-Learning-System-Based Model-Free Adaptive Predictive Control for Nonlinear MASs Under DoS Attacks. *IEEE/CAA J. Autom. Sin.* **2025**, *12*, 381–393.
15. Cai, G.; Yin, G.; Liu, Y.; et al. Stochastic Cooperative Adaptive Cruise Control With Sensor Data Distortion and Communication Delay. *IEEE Trans. Intell. Transp. Syst.* **2025**, *26*, 9500–9515.
16. Zhang, J.; Wang, Z.; Zhang, H. Data-Based Optimal Control of Multiagent Systems: A Reinforcement Learning Design Approach. *IEEE Trans. Cybern.* **2019**, *49*, 4441–4449.
17. Feng, T.; Zhang, J.; Tong, Y.; et al. Q-Learning Algorithm in Solving Consensusability Problem of Discrete-Time Multi-Agent Systems. *Automatica* **2021**, *128*, 109576.
18. Chen, L.; Dong, C.; Dai, S.L. Adaptive Optimal Consensus Control of Multiagent Systems With Unknown Dynamics and Disturbances via Reinforcement Learning. *IEEE Trans. Artif. Intell.* **2024**, *5*, 2193–2203.
19. Niu, B.; Wang, X.A.; Wang, H.Q.; et al. Adaptive RL Optimized Bipartite Consensus Tracking for Heterogeneous Nonlinear MASs Under a Switching Threshold Event Triggered Strategy. *IEEE Trans. Autom. Sci. Eng.* **2024**, *21*, 7379–7389.
20. Lin, M.; Zhao, B.; Liu, D. Policy Gradient Adaptive Critic Designs for Model-Free Optimal Tracking Control With Experience Replay. *IEEE Trans. Syst. Man Cybern. Syst.* **2022**, *52*, 3692–3703.
21. Yang, X.; Zhang, H.; Wang, Z. Data-Based Optimal Consensus Control for Multiagent Systems With Policy Gradient Reinforcement Learning. *IEEE Trans. Neural Netw. Learn. Syst.* **2022**, *33*, 3872–3883.
22. Li, J.; Ji, L.; Zhang, C.; et al. Optimal Couple-Group Tracking Control for the Heterogeneous Multi-Agent Systems With Cooperative-Competitive Interactions via Reinforcement Learning Method. *Inf. Sci.* **2022**, *610*, 401–424.
23. Mohammadi, M.; Arefi, M.M.; Setoodeh, P.; et al. Optimal Tracking Control Based on Reinforcement Learning Value Iteration Algorithm for Time-Delayed Nonlinear Systems With External Disturbances and Input Constraints. *Inf. Sci.* **2021**, *554*, 84–98.

24. Ji, Y.; Wang, G.; Li, Q.; et al. Event-Triggered Optimal Consensus of Heterogeneous Nonlinear Multi-Agent Systems. *Mathematics* **2022**, *10*, 10105–10115.
25. Li, G.; Wang, L. Adaptive Output Consensus of Heterogeneous Nonlinear Multiagent Systems: A Distributed Dynamic Compensator Approach. *IEEE Trans. Autom. Control* **2023**, *68*, 2483–2489.
26. Liu, D.; Wei, Q. Policy Iteration Adaptive Dynamic Programming Algorithm for Discrete-Time Nonlinear Systems. *IEEE Trans. Neural Netw. Learn. Syst.* **2014**, *25*, 621–634.
27. Zhang, H.; Jiang, H.; Luo, Y.; et al. Data-Driven Optimal Consensus Control for Discrete-Time Multi-Agent Systems With Unknown Dynamics Using Reinforcement Learning Method. *IEEE Trans. Ind. Electron.* **2017**, *64*, 4091–4100.
28. Wen, G.; Yu, Y.; Peng, Z.; et al. Dynamical Group Consensus of Heterogenous Multi-Agent Systems With Input Time Delays. *Neurocomputing* **2016**, *175*, 278–286.
29. Guo, X.G.; Liu, P.M.; Wang, J.L.; et al. Event-Triggered Adaptive Fault-Tolerant Pinning Control for Cluster Consensus of Heterogeneous Nonlinear Multi-Agent Systems Under Aperiodic DoS Attacks. *IEEE Trans. Netw. Sci. Eng.* **2021**, *8*, 1941–1956.
30. Li, K.; Hua, C.; You, X.; et al. Output Feedback Predefined-Time Bipartite Consensus Control for High-Order Nonlinear Multiagent Systems. *IEEE Trans. Circuits Syst. I Regul. Pap.* **2021**, *68*, 3069–3078.
31. Li, X.; Yu, Z.; Li, Z.; et al. Group Consensus via Pinning Control for a Class of Heterogeneous Multi-Agent Systems With Input Constraints. *Inf. Sci.* **2021**, *542*, 247–262.
32. Zhao, G.; Hua, C. Leaderless and Leader-Following Bipartite Consensus of Multiagent Systems With Sampled and Delayed Information. *IEEE Trans. Neural Netw. Learn. Syst.* **2023**, *34*, 2220–2233.
33. Ai, X. Adaptive Robust Bipartite Consensus of High-Order Uncertain Multi-Agent Systems Over Cooperation-Competition Networks. *J. Frankl. Inst.* **2020**, *357*, 1813–1831.
34. Jiang, Y.; Ji, L.; Liu, Q.; Yang, S.; Liao, X. Couple-Group Consensus for Discrete-Time Heterogeneous Multiagent Systems With Cooperative–Competitive Interactions and Time Delays. *Neurocomputing* **2018**, *319*, 92–101.
35. Wen, G.; Li, B. Optimized Leader-Follower Consensus Control Using Reinforcement Learning for a Class of Second-Order Nonlinear Multiagent Systems. *IEEE Trans. Syst. Man, Cybern. Syst.* **2022**, *52*, 5546–5555.
36. Liu, C.L.; Liu, F. Dynamical Consensus Seeking of Heterogeneous Multi-Agent Systems Under Input Delays. *Int. J. Commun. Syst.* **2013**, *26*, 1243–1258.
37. Luo, B.; Liu, D.; Wu, H.N.; et al. Policy Gradient Adaptive Dynamic Programming for Data-Based Optimal Control. *IEEE Trans. Cybern.* **2017**, *47*, 3341–3354.
38. Peng, Z.; Hu, J.; Shi, K.; et al. A Novel Optimal Bipartite Consensus Control Scheme for Unknown Multi-Agent Systems via Model-Free Reinforcement Learning. *Appl. Math. Comput.* **2020**, *369*, 124821.
39. Ji, L.; Lin, Z.; Zhang, C.; et al. Data-Based Optimal Consensus Control for Multiagent Systems With Time Delays: Using Prioritized Experience Replay. *IEEE Trans. Syst. Man, Cybern. Syst.* **2024**, *54*, 3244–3256.