Cluster Synchronization for Discrete-Time Zero-Sum Graphical Games with Unknown Constrained-Input Systems

Zahra Jahan[1], Abbas Dideban[1•] and Farzaneh Abdollahi[2]

***Abstract*-- This paper addresses the synchronization issue of agents with their respective leaders in each cluster for unknown discrete-time zero-sum graphical games with constrained input. To solve the coupled Hamilton-Jacobi-Isaacs equations under the assumption of unknown dynamics, an adaptive optimal distributed technique based on value iteration heuristic dynamic programming is proposed. An actor-critic framework is employed to approximate the value functions, control policies, and worst-case disturbance policies necessary for implementing the proposed method. Additionally, neural network identifiers are utilized to determine each agent's unknown dynamics. To prevent system instability, a constraint on control inputs is incorporated into the design method. By considering disturbances in the dynamics, the proposed solutions are made robust against unpredictable events, enhancing performance and stability. Furthermore, the closed-loop system's stability is proven. Finally, the theoretical results are validated through simulation outcomes.**

***Index Terms*-- Cluster synchronization, Discrete-time graphical zero-sum games, External disturbances, Input constraint, Neural network, Reinforcement learning, Unknown dynamics.**

I. INTRODUCTION

Due to the wide range of distributed control applications in multi-agent systems (MAS) across various technical domains, research in these fields has attracted considerable attention over the last two decades [1, 2]. In recent years, distributed control methods have been developed to address consensus and synchronization problems [3, 4]. As more and more tasks are being assigned to MAS in various fields, it has become necessary in some cases to divide the network's agents into several subgroups so that they can perform different tasks in a coordinated manner. By dividing agents into these subgroups, known as clusters, MAS can effectively perform various tasks while maintaining coordination and achieving common goals. Indeed, the importance of cluster consensus/synchronization cannot be understated. Examples of applications for cluster consensus and synchronization can be found across a wide range of fields, including opinion formation, bacterial colony pattern formation, and many others.

In the cluster synchronization problem, agents in separate clusters achieve different states, while agents within the same cluster converge to the same state. Cluster consensus problems are generally categorized into two types: cluster tracking and cluster regulation. In cluster regulation, agents within a cluster converge to a similar value, which is often uncontrollable [5, 6]. Cluster tracking involves agents within the same cluster following a leader, although the paths taken by different leaders eventually diverge [7, 8].

The cluster consensus study of MAS using inter-cluster nonidentical inputs has been examined in [9]. Reference [10] addresses the issue of distributed feedback controllers for cluster consensus in generic linear MASs with a directed interaction topology. While these studies provide important insights, they do not consider performance index optimization or guarantee optimality in the proposed methods. To address this, game theory can be utilized as a framework for solving the problem.

Optimal multi-agent control problems, in which each agent seeks to optimize their performance index and obtain an optimal policy, are well-suited for research within the framework of game theory [11]. Since many real-world MAS problems involve external disturbances, solving multi-agent games with unknown external disturbances is crucial. Neglecting these factors can result in performance degradation and instability.

Graphical games, initially developed for continuous-time (CT) systems, are employed to optimally solve distributed leader-follower consensus problems in both linear [12–14] and nonlinear [15] systems. In these games, each follower's performance index, actions, and local error dynamics depend on local information from their neighbors. Zero-sum differential graphical games for CT systems, which account for the existence of external disturbances, have been explored in [16, 17]. Several studies have also investigated discrete-time (DT) two-player zero-sum games [18–20]. Additionally, linear N-player DT graphical games have been studied by [21, 22].

To determine the solutions of multi-agent graphical games that include external disturbances, the Hamilton-Jacobi-Isaacs (HJI) equations must be solved. However, solving these nonlinear equations can be challenging in certain cases. Consequently, approximate-based approaches are often used, typically employing reinforcement learning or other iterative methods.

In recent years, reinforcement learning algorithms [23] have gained traction as powerful tools for solving multi-agent games [24, 25]. Two well-known reinforcement learning methods are policy iteration (PI) and value iteration (VI), both of which can be implemented online using a critic-actor structure. In this framework, the critic is a neural network that estimates the value function, while the actor estimates the optimal policies.

In practice, physical systems often exhibit complex dynamics that are challenging to model accurately. Additionally, the presence of saturation nonlinearity in many

1 . Electrical Engineering Department, Semnan University, Semnan, Iran
• Corresponding author Email: adideban@semnan.ac.ir
2 . Electrical Engineering Department, Amirkabir University, Tehran, Iran

actuators requires careful consideration in controller design. Neglecting saturation nonlinearity can result in performance degradation or even instability, as demonstrated in numerous studies [26, 27]. Therefore, solving multi-agent games with unknown dynamics and constrained input is essential for maintaining stability and achieving optimal performance.

Many studies on cluster consensus have been conducted, but the methods presented have not addressed the optimality problem [5–10]. While numerous references have tackled the graphical game problem, they have not considered cluster synchronization [12–17, 21, 22]. References [28] and [29] address the cluster synchronization problem in graphical games; however, they do not account for disturbances in the dynamics of agents. In our proposed method, we tackle cluster synchronization for zero-sum graphical games while considering disturbances in the dynamics of agents. Considering disturbances in the dynamics helps ensure that the proposed solutions are robust and can withstand unpredictable events, ultimately leading to improved performance and stability.

To the best of our knowledge, no result has been reported on cluster synchronization for DT zero-sum graphical games with unknown constrained-input systems that can address the N-player optimal leader-follower cluster consensus problem. So this paper first introduces multi-agent DT linear zero-sum graphical games with unknown constrained-input systems and external disturbances, and then cluster synchronization is proposed for these games.

The following are the main contributions of this paper:

• The first contribution is the introduction of DT zero-sum graphical games for linear MASs that utilize the local information of neighbor agents to achieve the optimal leader-follower synchronization problem.

• For the first time, the cluster synchronization of DT zero-sum graphical games is introduced.

• A VI HDP algorithm in an online and distributed fashion is proposed.

• The presented algorithm solves the games under the assumption of unknown dynamics, where each agent's unknown dynamics are identified using an identifier.

• Constraints on control inputs and external disturbances are considered to make the proposed algorithm more applicable to real-world problems.

Structure: This paper is organized into the following sections: Section II provides background information on graphs, the cluster synchronization problem, and optimal distributed consensus control for DT MAS with external disturbances. In Section III, the focus shifts to linear zero-sum DT graphical games with disturbances and control constraints, including a proof of closed-loop stability. The proposed optimal distributed algorithm is presented afterward in Section IV. Section V describes the identifier used for each agent's unknown dynamics, while Section VI introduces the actor-critic structure employed in the proposed algorithm. Finally, the simulation results are presented, followed by the conclusions.

II. PRELIMINARIES

This section begins with an overview of graph theory, followed by a review of the cluster synchronization problem and optimal distributed consensus control for DT MAS with external disturbances.

A. Graphs

The directed graph $Gr=(P,\Sigma)$ provides a description of the interactive topology for N agents' information exchange, where a set of graph edges is $\sum \subset P\times P$ and a set of graph nodes is $P=\{p_0,.....,p_N\}$. $C=[c_{ij}]\in R^{N\times N}$ represents an adjacency matrix for the graph such that $c_{ij}>0$ if $(p_j,p_i)\in\sum$, otherwise $c_{ij}=0$ where (p_j,p_i) means that the agent i can get information from the agent j but not necessarily vice versa. The list of the node p_i's neighbors is shown with $N_i=\{p_j:(p_j,p_i)\in\sum\}$. *Denotes* the in-degree matrix of Gr with $q_i=\sum_{j\in N_i}c_{ij}$. The Laplacian matrix of the graph is indicated by $L=Q-C$. The pinning matrix is given by $G=diag\{g_i\}\in R^{N\times N}$, where $g_i\geq 0$ is the pinning gain. If the agent i is connected to the leader, it is non-zero; otherwise, it is equal to zero. A graph has a spanning tree if there is a directed path from an agent called the root to all other agents. This paper assumes that the graph has a spanning tree.

B. Problem Formulation

On the graph Gr with N follower agents, the i^{th} agent's dynamic is as follows

$$s_i(t+1)=\mathrm{A}s_i(t)+\mathrm{B}_i u_i(t)+E_i\omega_i(t) \qquad (1)$$

where $s_i(t)\in\mathbb{R}^n$, $u_i(t)\in\mathbb{R}^m$ and $\omega_i(t)\in\mathbb{R}^d$ are the state, control input, and external disturbance vector of agent i, respectively. A, B_i and E_i are the state, control input, and disturbance matrices, respectively, which are all considered unknown in our studies. The assumption is that there are a clusters and a virtual leaders. In each cluster, the agents must track the corresponding leader.

The leader dynamics for each cluster $s_k(t)\in\mathbb{R}^n$ are described as follows

$$s_k(t+1)=\mathrm{A}s_k(t) \qquad k=01,....,0a \qquad (2)$$

Where, in each cluster, the leader is at least connected to one of the follower nodes.

Synchronizing all follower agents' states to the leader in each cluster is the goal of the cluster leader-follower consensus. Cluster synchronization for the MAS (1) is achieved when, for each agent, there is a control policy u_i that guarantees the following conditions for any initial state $s_i(0)$

$$\lim_{t\to\infty}\left\|s_i(t)-s_j(t)\right\|=0 \quad \forall i,j=1,...,N \quad \bar{i}=\bar{j} \qquad (3)$$

$$\lim_{t\to\infty}\left\|s_i(t)-s_j(t)\right\|\neq 0 \quad \forall i,j=1,...,N \quad \bar{i}\neq\bar{j} \qquad (4)$$

Which is cluster related to the agent i and $\bar{j}$ is cluster related to the agent j.

Equation (3) shows the synchronization of agents inside a cluster. So that the agents in each cluster follow the leader of the cluster. Therefore, (3) can be rewritten as below:

$$\lim_{t\to\infty}\left\|s_i(t)-s_k(t)\right\|=0 \qquad \forall i=1,...,N \tag{5}$$

cluster $\bar{i}$'s leader is represented by s_k.

Equation (4) shows that by choosing a different initial value in equation (2), different paths are created for the leaders:

$$\lim_{t\to\infty}\left\|s_r(t)-s_e(t)\right\|\neq 0 \quad \forall r,e=01,...,0a \quad r\neq e \tag{6}$$

The cluster local error of agent i [30] is

$$\rho_i(t)=\sum_{j\in N_i}c_{ij}\left(s_j(t)-s_i(t)\right)+g_i\left(s_t(t)-s_i(t)\right) \tag{7}$$

The network cluster local error for all agents is

$$\rho(t)=-\left((L+G)\otimes I_n\right)(s(t)-\bar{s}_t(t)) \tag{8}$$

where $s=\left[s_1^T,\ldots,s_N^T\right]^T\in\mathbb{R}^{nN}$, $\rho=\left[\rho_1^T,\ldots,\rho_N^T\right]^T\in\mathbb{R}^{nN}$, $\bar{s}_t(t)=\left[s_t^T(t),\ldots,s_t^T(t)\right]^T\in\mathbb{R}^{nN}$.

For cluster synchronization, the error vector is specified as

$$\theta(t)=s(t)-\bar{s}_t(t)\in\mathbb{R}^{nN} \tag{9}$$

If in each cluster a root node is connected to the leader and the graph has a spanning tree, $(L+G)$ is non-singular [30]. It is demonstrated in [21] that if $(L+G)$ is non-singular, the synchronization error vector is bounded as

$$\left\|\theta(t)\right\|\leq\left\|\rho(t)\right\|/\underline{\sigma}(L+G) \tag{10}$$

Where is a matrix's smallest singular value. Therefore, cluster leader-follower synchronization can be achieved by keeping the cluster local error small.

For simplicity, $s_i(t)$ is written as s_{it} from now on, and other variables are considered similarly.

By using equations (1) and (7), the i^{th} agent's cluster local error dynamics are obtained as

$$\rho_{i(t+1)}=A\rho_{it}-(q_i+g_i)(B_iu_{it}+E_i\omega_{it}) \tag{11}$$

$$+\sum_{j\in N_i}c_{ij}(B_ju_{jt}+E_j\omega_{jt})$$

To obtain cluster leader-follower synchronization, the development of a distributed controller for agents in each cluster is suggested to minimize equation (11) for $\omega_{it}\neq 0$, under the unknown dynamics of the system.

III. Multi-player zero-sum discrete-time graphical game

In this section, we introduce a novel type of game, called linear zero-sum DT graphical games, which considers both disturbance and control constraints. By utilizing the cluster local error dynamics (11) and introducing a local performance index, these games are defined.

The i^{th} agent's local performance index is defined as:

$$J_i(\rho_{it},u_{it},u_{-it},\omega_{it},\omega_{-it})=\sum_{t=0}^{\infty}U_i(\rho_{it},u_{it},u_{-it},\omega_{it},\omega_{-it}) \tag{12}$$

$$=\frac{1}{2}\sum_{t=0}^{\infty}\rho_{it}^TO_{ii}\rho_{it}+W(u_{it})+\sum_{j\in N_i}W(u_{jt})$$

$$-\gamma^2\omega_{it}^TT_{ii}\omega_{it}-\gamma^2\sum_{j\in N_i}\omega_{jt}^TT_{ij}\omega_{jt}$$

Where $O_{ii}>0\in\mathbb{R}^{n\times n}$, $T_{ii}>0\in\mathbb{R}^{d\times d}$, $\gamma>0$ is a prescribed constant and $W(.)>0$. The control input and disturbance of i^{th} agent's neighbors are denoted by $u_{-i}=\{u_j\mid j\in N_i\}$ and $\omega_{-i}=\{\omega_j\mid j\in N_i\}$, respectively. For each player, a nonquadratic functional [31] is used to take into account the control input constraints:

$$W(u_{it})=2\int_0^{u_i}\phi^{-T}(\bar{Y}^{-1}x)\bar{Y}ydx \tag{13}$$

Where $y>0$ i a diagonal positive definite matrix, $x\in R^m$, $\phi\in R^m$, $\phi^{-1}(u_{it})=[\psi^{-1}(u_{it}^1)\psi^{-1}(u_{it}^2)....\psi^{-1}(u_{it}^m)]^T$ where u_{it}^z is the z-th element of the vector u_{it}, $z=0,...,m$. $\psi(.)$ is a monotonic odd bounded function satisfying $|\psi(.)|\leq 1$ and its first derivative is bounded. $\bar{Y}$ is a bound for actuators. In this paper, $\psi(.)=\tanh(.)$.

Each i^{th} agent 's value function is defined as:

$$V_i(\rho_{it})=\sum_{b=t}^{\infty}U_i(\rho_{ib},u_{ib},u_{-ib},\omega_{ib},\omega_{-ib}) \tag{14}$$

A. Bounded L2-gain synchronization problem

For zero-sum DT graphical games, it is desirable to find a constrained control input u_{it} that solves the synchronization problem when $\omega_{it}=0$. This input should satisfy the following bounded L_2 -gain condition for a given $\gamma>\gamma^*$, with $\omega_{it}\neq 0$ for all players.

$$\sum_{t=0}^{M}(\rho_{it}^TO_{ii}\rho_{it}+W(u_{it})+\sum_{j\in N_i}W(u_{jt})) \tag{15}$$

$$\leq\gamma^2\sum_{t=0}^{M}(\omega_{it}^TT_{ii}\omega_{it}+\sum_{j\in N_i}\omega_{jt}^TT_{ij}\omega_{jt})+\beta(\varepsilon_{i0})$$

For several bounded functions β such that $\beta(0)=0$. Let γ^* is the minimum amount of γ that satisfies the above bounded L_2 -gain condition.

B. Coupled Hamilton-Jacobi-Isaacs equation

Based on equations (11) and (12), the Hamiltonian function for each agent can be defined as follows:

$$H_i\left(\rho_{it},\nabla V_i\left(\rho_{i(t+1)}\right),u_{it},u_{-it},\omega_{it},\omega_{-it}\right)= \quad (16)$$
$$\nabla V_i\left(\rho_{i(t+1)}\right)^T\begin{pmatrix}A\rho_{it}-(q_i+g_i)(B_iu_{it}+E_i\omega_{it})\\+\sum_{j\in N_i}c_{ij}(B_ju_{jt}+E_j\omega_{jt})\end{pmatrix}$$
$$+\frac{1}{2}\begin{pmatrix}\rho_{it}^TO_{ii}\rho_{it}+W\left(u_{it}\right)+\sum_{j\in N_j}W\left(u_{jt}\right)\\-\gamma^2\omega_{it}^TT_{ii}\omega_{it}-\gamma^2\sum_{j\in N_i}\omega_{jt}^TT_{ij}\omega_{jt}\end{pmatrix}=0,V_i(0)=0$$

By employing the stationarity conditions, $\frac{\partial H_i}{\partial u_{it}}=0$ and $\frac{\partial H_i}{\partial \omega_{it}}=0$, the bounded optimal control and disturbance policies are determined in the following ways:

$$u_{it}^*=\underset{u_{it}}{\operatorname{argmin}}(H_i(\rho_{it},\nabla V_i(\rho_{i(t+1)}),u_{it},u_{-it},\omega_{it},\omega_{-it})) \quad (17)$$
$$=\bar{Y}\phi\left((\bar{Y}y)^{-1}(q_i+g_i)B_i^T\nabla V_i^*(\rho_{i(t+1)})\right)$$
$$\omega_{it}^*=\underset{\omega_{it}}{\operatorname{argmax}}(H_i(\rho_{it},\nabla V_i(\rho_{i(t+1)}),u_{it},u_{-it},\omega_{it},\omega_{-it})) \quad (18)$$
$$=-\frac{1}{\gamma^2}(q_i+g_i)T_{ii}^{-1}E_i^T\nabla V_i^*(\rho_{i(t+1)})$$

By substituting equations (17) and (18) into equation (16), we obtain the following coupled DT HJI equations:

$$H_i\left(\rho_{it},\nabla V_i\left(\varepsilon_{i(t+1)}\right),u_{it}^*,u_{-it}^*,\omega_{it}^*,\omega_{-it}^*\right)= \quad (19)$$
$$\nabla V_i\left(\rho_{i(t+1)}\right)^T\begin{pmatrix}A\rho_{it}-\left(q_i+g_i\right)\begin{pmatrix}\bar{Y}B_i\phi\begin{pmatrix}\left(\bar{Y}y\right)^{-1}\left(q_i+g_i\right)\\\times B_i^T\nabla V_i^*\left(\rho_{i(t+1)}\right)\end{pmatrix}\\\begin{pmatrix}-\frac{E_i}{\gamma^2}(q_i+g_i)T_{ii}^{-1}\\\times E_i^T\nabla V_i^*(\rho_{i(t+1)})\end{pmatrix}\end{pmatrix}\\+\sum_{j\in N_i}c_{ij}\begin{bmatrix}\bar{Y}B_j\phi\begin{pmatrix}\left(\bar{Y}R\right)^{-1}\left(q_j+g_j\right)\\\times B_j^T\nabla V_j^*\left(\rho_{j(t+1)}\right)\end{pmatrix}\\\begin{pmatrix}-\frac{E_j}{\gamma^2}(q_j+g_j)T_{jj}^{-1}\\\times E_j^T\nabla V_j^*(\rho_{j(t+1)})\end{pmatrix}\end{bmatrix}\end{pmatrix}$$
$$+\frac{1}{2}\begin{pmatrix}\rho_{it}^TO_{ii}\rho_{it}+W\left(u_{it}^*\right)+\sum_{j\in N_j}W\left(u_{jt}^*\right)\\-\frac{1}{\gamma^2}\left(q_i+g_i\right)^2\nabla V_i^*\left(\rho_{i(t+1)}\right)^TE_iT_{ii}^{-1}E_i^T\nabla V_i^*\left(\rho_{i(t+1)}\right)\\-\frac{1}{\gamma^2}\sum_{j\in N_i}\left(q_j+g_j\right)^2\nabla V_j^*\left(\rho_{j(t+1)}\right)^T\\\times B_jT_{jj}^{-1}T_{ij}T_{jj}^{-1}B_j^T\nabla V_j^*\left(\rho_{j(t+1)}\right)\end{pmatrix}=0,V_i(0)=0$$

Solving the coupled DT HJI equations can be a difficult task, often leading to intractable results. To overcome this challenge, we propose the use of the VI algorithm as an approximate solution method for these equations.

Theorem 1. In each cluster, let $V_i^*(\rho_{it})<0$ satisfies equation (19). Let the condition

$$(\rho_{it}^TQ_{ii}\rho_{it}+W(u_{it})+\sum_{j\in N_i}W(u_{jt})$$
$$\le\gamma^2(\omega_{it}^TT_{ii}\omega_{it}+\sum_{j\in N_i}w_{jt}^TT_{ij}\omega_{jt})$$

holds. Assuming control and disturbance policies are respectively given by (17) and (18) in terms of V_i^*, and that the communication graph has a spanning tree with $g_i\neq 0$ for at least one agent, then the dynamics of the cluster local error (11) will be asymptotically stable, resulting in synchronization of all agents' states within each cluster to the leader state.

Proof. Considering the first difference of equation (14), the Bellman equation for each agent i is obtained as follows

$$V_i(\rho_{it})=U_i(\rho_{it},u_{it},u_{-it},\omega_{it},\omega_{-it})+V_i(\rho_{i(t+1)}) \quad (20)$$

Consider $-V_i^*(\rho_{it})>0$ as Lyapunov function for (11), and from (20) we have

$$-(V_i^*(\rho_{i(t+1)})-V_i^*(\rho_{it})) \quad (21)$$
$$=U_i^*(\rho_{it},u_{it}^*,u_{-it}^*,\omega_{it}^*,\omega_{-it}^*)<0$$

Therefore, the dynamics of cluster local error (11) will be asymptotically stable, leading to the synchronization of all agent states to their respective leader states.

IV. Value Iteration Algorithm for DT Zero-Sum Graphical Games

This section introduces an online VI HDP method for solving the DT zero-sum graphical game within each cluster based on Bellman equations (20). Algorithm 1 is applied to all clusters, allowing for the coupled Bellman equations of each cluster to be solved and optimal values, control policies, and disturbance policies to be obtained in each cluster.

Algorithm 1

Value Iteration Algorithm for DT Zero-Sum Graphical Games

(Initialization). Give arbitrary initial control and disturbance policies and values for all agents.

(Policy evaluation). Solve the following equation to obtain V_i^{l+1}

$$V_i^{l+1}(\rho_{it})=U_i(\rho_{it},u_{it}^l,u_{-it}^l,\omega_{it}^l,\omega_{-it}^l)+V_i^l(\rho_{i(t+1)}) \quad (22)$$

(Policy improvement). Update the disturbance and control policies with the following equations.

$$u_{it}^{l+1}=\bar{Y}\phi\left(\left(\bar{Y}y\right)^{-1}\left(q_i+g_i\right)B_i^T\nabla V_i\left(\rho_{i(t+1)}\right)^{l+1}\right) \quad (23)$$

$$\omega_{it}^{l+1}=-\frac{1}{\gamma^2}(q_i+g_i)T_{ii}^{-1}E_i^T\nabla V_i(\rho_{i(t+1)})^{l+1} \quad (24)$$

For all i, when $\left\|V_i^{l+1}(\rho_{it})-V_i^l(\rho_{it})\right\|\le\delta$ end. δ is a small constant.

l denotes the iteration index.

V. Neural Network-Based System Approximation

This study assumes that the drift, input, and disturbance dynamics of each agent in all clusters are unknown. To address this challenge, we employ a neural network-based identification technique that approximates the unknown system dynamics as follows:

$$s_i(t+1) = W_{is}^T \varphi_{is}(v_{is}^T z_{is}(t)) + \varepsilon_{is}(t) \tag{25}$$

where $\varphi_{is}(.)$ is the activation function and it is presumed to be bounded, $\varepsilon_{is}(t)$ is the NN estimation error, $z_{is}(t) = [s_{it}^T \quad u_{it}^T \quad \omega_{it}^T]^T \in R^D$ is the NN input with $D = n+m+d$, v_{is}^T and W_{is}^T denote the ideal weight matrix between the input layer and the hidden layer, the hidden layer and the output layer, respectively.

In the system identification process, v_{is}^T is assumed to be constant and only W_{is}^T is adjusted. Hence, the identifier network output is described as

$$\hat{s}_i(t+1) = \hat{W}_{is}^T \varphi_{is}(Z_{is}(t)) \tag{26}$$

where $Z_{is}(t) = v_{is}^T z_{is}(t)$, $\hat{s}_i$ is the estimated state vector and $\hat{W}_{is}$ is the estimation of W_{is}.

The approximation error for system identification is defined as

$$\begin{aligned} e_{is} &= \hat{s}_i(t+1) - s_i(t+1) \\ &= \hat{W}_{is}^T \varphi_{is}(Z_{is}(t)) - s_i(t+1) \end{aligned} \tag{27}$$

The squared approximation error is defined as follows

$$E_{is} = \frac{1}{2}(e_{is})^T e_{is} \tag{28}$$

The identifier weights are updated using the gradient descent rule as

$$\begin{aligned} \hat{W}_{is}^{(l+1)T} &= \hat{W}_{is}^{lT} \\ &-\mu_{is}\varphi_{is}(Z_{is}(t))(\hat{W}_{is}^{lT}\varphi_{is}(Z_{is}(t)) - s_i^l(t+1))^T \end{aligned} \tag{29}$$

where $0 < \mu_{is} < 1$ denotes the learning rate of the identifier network.

After the learning process in the NN, the identifier weight matrix will converge to a certain value $\hat{W}_{ism}^T$. Then, the identifier network output is expressed as

$$\begin{aligned} \hat{s}_i(t+1) &= \hat{A}s_i(t) + \hat{B}u_i(t) + \hat{E}\omega_i(t) \\ &= \hat{W}_{ism}^T \varphi_{is}(v_{is}^T z_{is}(t)) \end{aligned} \tag{30}$$

where $\hat{A}$, $\hat{B}$ and $\hat{E}$ are the estimations of A, B and E, respectively.

VI. Implementation of the value iteration HDP algorithm for unknown DT zero-sum graphical games

This section provides an actor-critic structure for the implementation of the HDP algorithm in each cluster. For all agents within each cluster, the critic network is built to estimate the optimal value function and carry out the policy assessment (22). The actor approximators are constructed to perform the policy improvement equations (23) and (24), which estimate the bounded optimal control and disturbance policies for agents in each cluster. Additionally, each agent's unknown dynamics are approximated using a neural network.

A. Actor-critic approximators and tuning

In each cluster, for any agent i, the control and disturbance policies are estimated using actor approximators $\hat{u}_i(.|\hat{W}_{ia})$ and $\hat{\omega}_i(.|\tilde{W}_{id})$, respectively. Also, the optimal value function is estimated using the critic approximator $\hat{V}_i(.|\hat{W}_{ic})$ so that

$$\hat{u}_{ik}(\hat{W}_{ia}) = \hat{W}_{ia}^T \varphi_{iu}(t) \tag{31}$$

$$\hat{\omega}_{ik}(\hat{W}_{id}) = \hat{W}_{id}^T \varphi_{id}(t) \tag{32}$$

$$\hat{V}_{ik}(\hat{W}_{ic}) = \varphi_{ic}(t)^T \hat{W}_{ic}^T \varphi_{ic}(t) \tag{33}$$

where $\hat{W}_{ia}$, $\hat{W}_{id}$ are the estimated actor weights for control and disturbance, respectively and $\hat{W}_{ic}$ is the estimated critic weight. φ_{iu}, φ_{id} and φ_{ic} are the activation functions for control actor, disturbance actor, and critic, respectively. It is assumed that all activation functions are bounded, i.e., $\|\varphi_{iu}(.)\| \le \varphi_{ium}$, $\|\varphi_{id}(.)\| \le \varphi_{idm}$ and $\|\varphi_{ic}(.)\| \le \varphi_{icm}$. Each agent's activation function is equal to the cluster local error vector of that agent and its neighbor.

The actor network's approximation error for the control input is described as

$$e_{ia} = \hat{u}_{it}(\hat{W}_{ia}) - \tilde{u}_{it} = \hat{W}_{ia}^T \varphi_{iu}(t) - \tilde{u}_{it} \tag{34}$$

The control input $\tilde{u}_{ik}$ is given as

$$\tilde{u}_{it} = \bar{Y}\varphi\left(\left(\bar{Y}y\right)^{-1}(q_i + g_i)\hat{B}_i^T \nabla \hat{V}_i\left(\rho_{i(t+1)}\right)\right) \tag{35}$$

By using (33), (35) can be written as follows

$$\tilde{u}_{it} = \bar{Y}\varphi\left(\left(\bar{Y}y\right)^{-1}(q_i + g_i)\hat{B}_i^T F_i \hat{W}_{ic}^T\right) \tag{36}$$

where $F_i = [0....[I]_{ii}....0] \in \square^{n \times nN}$.

The following is the definition of the actor network's squared approximation error for control input

$$E_{ia} = \frac{1}{2}(e_{ia})^T e_{ia} \tag{37}$$

The actor weights for control input are updated using the gradient descent algorithm.

$$\hat{W}_{ia}^{(1+l)T} = \hat{W}_{ia}^{lT} - \mu_{ia}\left(\hat{W}_{ia}^{lT}\varphi_{iu}(t) - \tilde{u}_{it}^{l}\right)\left(\varphi_{iu}(t)\right)^{T} \tag{38}$$

$0 < \mu_{ia} < 1$ is the learning rate of the actor network for control input.

Similarly, the actor network's approximation error for disturbance is described as

$$e_{id} = \hat{\omega}_{ik}\left(\hat{W}_{id}\right) - \tilde{\omega}_{it} = \hat{W}_{id}^{T}\varphi_{id}(t) - \tilde{\omega}_{it} \tag{39}$$

The disturbance $\tilde{\omega}_{ik}$ can be defined as

$$\tilde{\omega}_{it} = -\frac{1}{\gamma^{2}}(q_i + g_i)T_{ii}^{-1}\hat{E}_{i}^{T}\nabla\hat{V}_{i}(\rho_{i(t+1)}) \tag{40}$$

By using (33), (40) can be written as follows

$$\tilde{\omega}_{it} = -\frac{1}{\gamma^{2}}(q_i + g_i)T_{ii}^{-1}\hat{E}_{i}^{T}F_{i}\hat{W}_{ic}^{T} \tag{41}$$

The following defines the disturbance actor's squared approximation error

$$E_{id} = \frac{1}{2}\left(e_{id}\right)^{T} e_{id} \tag{42}$$

To update the actor weights for the disturbance, the gradient descent rule is employed as follows.

$$\hat{W}_{id}^{(1+l)T} = \hat{W}_{id}^{lT} - \mu_{id}\left(\hat{W}_{id}^{lT}\varphi_{id}(t) - \tilde{\omega}_{ik}^{l}\right)\left(\varphi_{id}(t)\right)^{T} \tag{43}$$

$0 < \mu_{id} < 1$ represents the disturbance's learning rate.

The objective value function $\tilde{V}_{ik}$ is given by

$$\tilde{V}_{it} = \frac{1}{2}\begin{pmatrix} \rho_i^{T}O_{ii}\rho_i + W\left(\hat{u}_{it}^{l}\right) + \sum_{j\in N_j} W\left(\hat{u}_{jt}^{l}\right) \\ -\gamma^{2}\hat{\omega}_{it}^{lT}T_{ii}\hat{\omega}_{it}^{l} - \gamma^{2}\sum_{j\in N_i}\hat{\omega}_{jt}^{lT}T_{ij}\hat{\omega}_{jt}^{l} \end{pmatrix} + \hat{V}_{i(t+1)}^{l} \tag{44}$$

The following is the critic network's approximation error.

$$e_{ic} = \tilde{V}_{it} - \hat{V}_{it}\left(\hat{\omega}_{ic}\right) \tag{45}$$

The following defines the squared approximation error for the critic:

$$E_{ic} = \frac{1}{2}\left(e_{ic}\right)^{T} e_{ic} \tag{46}$$

The critic weights are updated using the gradient descent method as follows.

$$\hat{W}_{ic}^{(1+l)T} = \hat{W}_{ic}^{lT} - \mu_{ic}\left(\varphi_{ic}(t)^{T}\hat{W}_{ic}^{lT}\varphi_{ic}(t) - \tilde{V}_{ik}\right)\varphi_{ic}(t)(\varphi_{ic}(t))^{T} \tag{47}$$

where $0 < \mu_{ic} < 1$ denotes the critic network learning rate.

In the following section, Algorithm 2 is presented for the online tuning of actor-critic network weights in unknown zero-sum discrete-time graphical games. Additionally, to enhance understanding of the proposed algorithm, the flowchart of Algorithm 2 is provided in the Appendix.

Algorithm 2

Actor-Critic Network Weights Online Tuning for Unknown DT Zero-Sum Graphical Games

1. Initialize the critic weights with zero, and the actors and identifiers weights randomly.

2. Initialize the initial state $x_i(0)$ and $x_t(0)$ for all leader agents randomly.

3. Do Loop (l iterations)

- Calculate the local tracking error ρ_{i0} on the system trajectory for all agents.

- Calculate the control policies $\hat{u}_{it}^{l}$ by equation (31)

- Calculate the disturbance policies $\hat{\omega}_{it}^{l}$ by equation (32)

- Calculate the estimated state $\hat{s}_{i(t+1)}^{l}$ by equation (26)

- Calculate the cluster local error $\rho_{i(t+1)}^{l}$ (11) using the estimated states

- Calculate the value function $\hat{V}_{i(t+1)}^{l}$ by equation (33)

- Update the critic weights

$$\hat{W}_{ic}^{(1+l)T} = \hat{W}_{ic}^{lT} - \mu_{ic}\left(\varphi_{ic}(t)^{T}\hat{W}_{ic}^{lT}\varphi_{ic}(t) - \tilde{V}_{ik}\right)\varphi_{ic}(t)(\varphi_{ic}(t))^{T}$$

where $\tilde{V}_{ik}$ is gained by equation (44)

- Update the actor weights

$$\hat{W}_{ia}^{(1+l)T} = \hat{W}_{ia}^{lT} - \mu_{ia}\left(\hat{W}_{ia}^{lT}\varphi_{iu}(t) - \tilde{u}_{ik}^{l}\right)\varphi_{iu}(t)^{T}$$

$$\hat{W}_{id}^{(1+l)T} = \hat{W}_{id}^{lT} - \mu_{id}\left(\hat{W}_{id}^{lT}\varphi_{id}(t) - \tilde{\omega}_{ik}^{l}\right)(\varphi_{id}(t))^{T}$$

- Update the identifier weights

$$\hat{W}_{is}^{(l+1)T} = \hat{W}_{is}^{lT} - \mu_{is}\varphi_{is}(t)(\hat{W}_{is}^{lT}\varphi_{is}(t) - s_{i}^{l}(t+1))^{T}$$

- For all i, when $\left\|\hat{V}_{i}^{l+1}(\rho_{it}) - \hat{V}_{i}^{l}(\rho_{it})\right\| \le \delta$ end, where δ is a small constant.

Theorem 2. Let the weight updates for the identifier, control actor, disturbance actor, and critic NNs be given by (29), (38), (43), and (47), respectively. If the learning rates for the NNs are chosen appropriately, the weight estimation errors for the identifier, control actor, disturbance actor, and critic NNs will be uniformly ultimately bounded.

Proof. The errors in weight estimate are as $\tilde{W}_{is}^{l} = \hat{W}_{is}^{l} - W_{is}$, $\tilde{W}_{ia}^{l} = \hat{W}_{ia}^{l} - W_{ia}$, $\tilde{W}_{id}^{l} = \hat{W}_{id}^{l} - W_{id}$ and $\tilde{W}_{ic}^{l} = \hat{W}_{ic}^{l} - W_{ic}$ where W_{is}, W_{ia}, W_{id} and W_{ic} are ideal values of the identifier, control actor, disturbance actor, and critic network weights, respectively. Then, based on (29), (38), (43) and (47), one has

$$\tilde{W}_{is}^{(l+1)T} = \tilde{W}_{is}^{lT} - \mu_{is}(e_{is}^{l})\varphi_{is}{}^{T} \tag{48}$$

$$\tilde{W}_{ia}^{(l+1)T} = \tilde{W}_{ia}^{lT} - \mu_{ia}(e_{ia}^{l})\varphi_{iu}{}^{T} \tag{49}$$

$$\tilde{W}_{id}^{(l+1)T} = \tilde{W}_{id}^{lT} - \mu_{id}(e_{id}^{l})\varphi_{id}^{T} \quad (50)$$

$$\tilde{W}_{ic}^{(l+1)T} = \tilde{W}_{ic}^{lT} - \mu_{ic}(e_{ic}^{l})\varphi_{ic}^{T} \quad (51)$$

where $e_{ic}^{l} = \tilde{W}_{ic}^{T}\varphi_{ic}(t)$, $e_{ia}^{l} = \tilde{W}_{ia}^{T}\varphi_{iu}(t)$, $e_{id}^{l} = \tilde{W}_{id}^{T}\varphi_{id}(t)$ and $e_{is}^{l} = \tilde{W}_{is}^{T}\varphi_{is}(t)$.

The Lyapunov function candidate is defined as follows.

$$\begin{aligned}\Delta P_i(\tilde{W}_{ia}^{lT}, \tilde{W}_{id}^{lT}, \tilde{W}_{ic}^{lT}, \tilde{W}_{is}^{lT}) &= tr\{\tilde{W}_{ia}^{(l+1)T}\tilde{W}_{ia}^{(l+1)} - \tilde{W}_{ia}^{lT}\tilde{W}_{ia}^{l}\} \\ &+ tr\{\tilde{W}_{id}^{(l+1)T}\tilde{W}_{id}^{(l+1)} - \tilde{W}_{id}^{lT}\tilde{W}_{id}^{l}\} \\ &+ tr\{\tilde{W}_{ic}^{(l+1)T}\tilde{W}_{ic}^{(l+1)} - \tilde{W}_{ic}^{lT}\tilde{W}_{ic}^{l}\} \\ &+ tr\{\tilde{W}_{is}^{(l+1)T}\tilde{W}_{is}^{(l+1)} - \tilde{W}_{is}^{lT}\tilde{W}_{is}^{l}\} \\ &\le \mu_{ia}\|e_{ia}^{l}\|^2(\mu_{ia}\|\varphi_{iu}\|^2 - 2) \\ &\le \mu_{id}\|e_{id}^{l}\|^2(\mu_{id}\|\varphi_{id}\|^2 - 2) \\ &\le \mu_{ic}\|e_{ic}^{l}\|^2(\mu_{ic}\|\varphi_{ic}\|^2 - 2) \\ &\le \mu_{is}\|e_{is}^{l}\|^2(\mu_{is}\|\varphi_{is}\|^2 - 2)\end{aligned}$$

Since all activation functions are bounded, if $\mu_{ia} \le 2/\|\varphi_{iu}\|^2$, $\mu_{id} \le 2/\|\varphi_{id}\|^2$, $\mu_{ic} \le 2/\|\varphi_{ic}\|^2$ and $\mu_{is} \le 2/\|\varphi_{is}\|^2$, then $\Delta P_i(\tilde{W}_{ia}^{lT}, \tilde{W}_{id}^{lT}, \tilde{W}_{ic}^{lT}, \tilde{W}_{is}^{lT}) \le 0$.

So, the proof is complete.

VII. Simulation Study

This section provides an example of cluster synchronization in discrete-time zero-sum graphical games, demonstrating the effectiveness of the presented method in synchronizing agents with the leader in each cluster. As shown in Fig. 1, consider a MAS with eight agents. Agents 1 through 4 are located in the first cluster, where agent 4 is connected to the leader. Agents 5, 6, 7, and 8 are situated in the second cluster, with agent 8 connected to the leader.

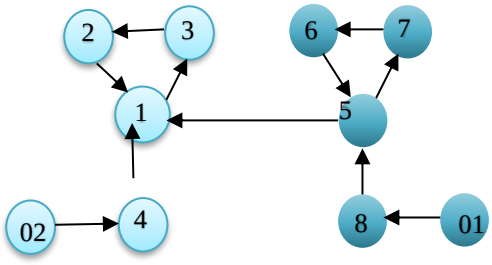

Fig. 1. Graph structure of multi-agent system with 8 agents

The drift, input, and disturbance matrices for each agent and leader are given as follows:

$$s_i(t+1) = As_i(t) + B_i u_i(t) + E_i\omega_i(t), \quad s_0(n+1) = As_0$$

$$s_i = \begin{bmatrix} s_{i1} \\ s_{i2} \end{bmatrix}, i = 01,02,1,2,3,4,5,6,7,8$$

$$A = \begin{pmatrix} 0.995 & 0.09983 \\ -0.09983 & 0.995 \end{pmatrix}$$

The drift matrix of the leader is as

$$B_1 = \begin{bmatrix} 0.2047 \\ 0.8984 \end{bmatrix} = B_2 = \begin{bmatrix} 0.2147 \\ 0.2895 \end{bmatrix} = B_3 = \begin{bmatrix} 0.02097 \\ 0.1897 \end{bmatrix}$$

$$B_4 = \begin{bmatrix} 0.2 \\ 0.01 \end{bmatrix} = B_5 = \begin{bmatrix} 0.3 \\ 0.9 \end{bmatrix} = B_6 = \begin{bmatrix} 0.2 \\ 0.3 \end{bmatrix} = B_7 = \begin{bmatrix} 0.09 \\ 0.1 \end{bmatrix}$$

$$= B_8 = \begin{bmatrix} 0.2 \\ 0.1 \end{bmatrix}$$

$$E_1 = \begin{bmatrix} 0.2047 \\ 0.8984 \end{bmatrix} = E_2 = \begin{bmatrix} 0.2147 \\ 0.2895 \end{bmatrix} = E_3 = \begin{bmatrix} 0.02097 \\ -0.1897 \end{bmatrix}$$

$$E_4 = \begin{bmatrix} 0.2 \\ 0.01 \end{bmatrix} = E_5 = \begin{bmatrix} 0.3 \\ 0.9 \end{bmatrix} = E_6 = \begin{bmatrix} 0.2 \\ 0.3 \end{bmatrix} = E_7 = \begin{bmatrix} 0.09 \\ 0.1 \end{bmatrix}$$

$$= E_8 = \begin{bmatrix} 0.2 \\ 0.1 \end{bmatrix}$$

The pinning gains are $g_1 = g_2 = g_3 = g_5 = g_6 = g_7 = 0$, $g_4 = g_8 = 1$ and the edge weights are considered as $c_{12} = c_{58} = 0.7, c_{15} = 0.1, c_{31} = c_{67} = 0.6, c_{23} = 0.4$, $e_{14} = e_{56} = e_{75} = 0.8$. The learning rates are chosen as $\mu_{ic} = \mu_{ia} = \mu_{is} = \mu_{id} = 0.1$. The disturbance attenuation is given by $\gamma = 1.5$ and bound for actuators is considered as $\bar{U} = 1$. A hyperbolic tangent function $\psi(.) = \tanh(.)$ is considered for the input constraint.

The performance index's matrices are chosen as $O_{ii} = T_{ii} = I_{2\times 2}$ for all agents.

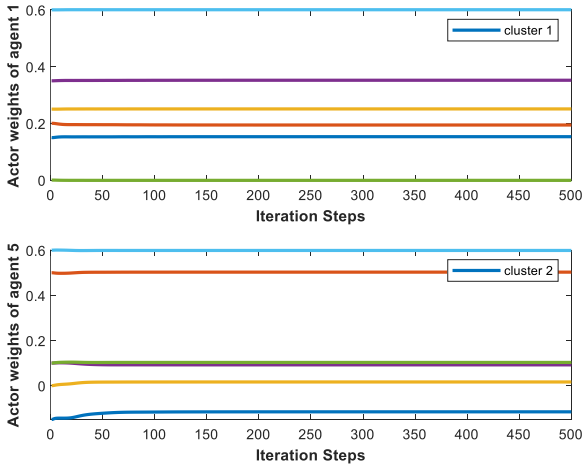

Fig. 2. The weights update of the control input actor for agent 1 and agent 5

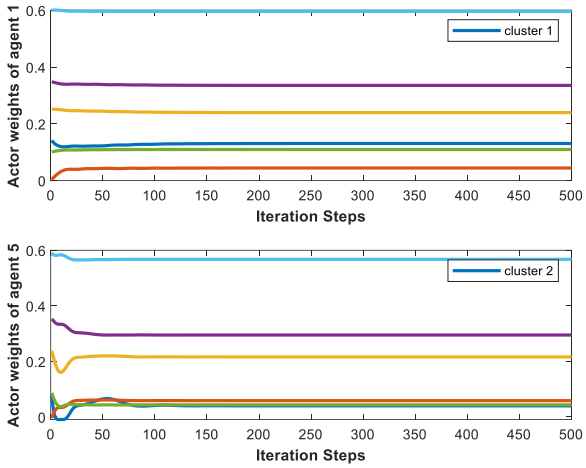

Fig. 3. The weights update of the disturbance actor for agent 1 and agent 5

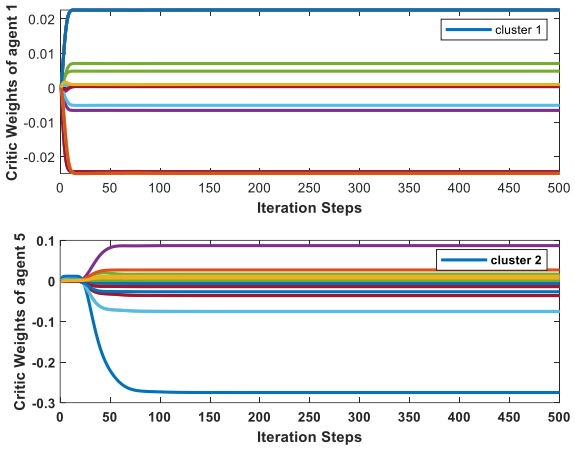

Fig. 4. Critic weights update for agent 1 and 5

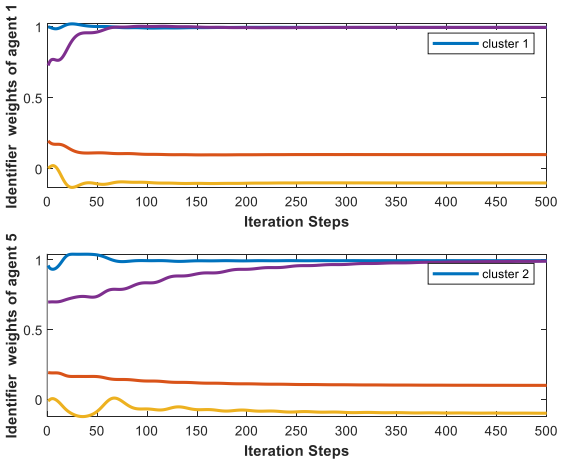

Fig. 5. Identifier weights update for agent 1 and 5

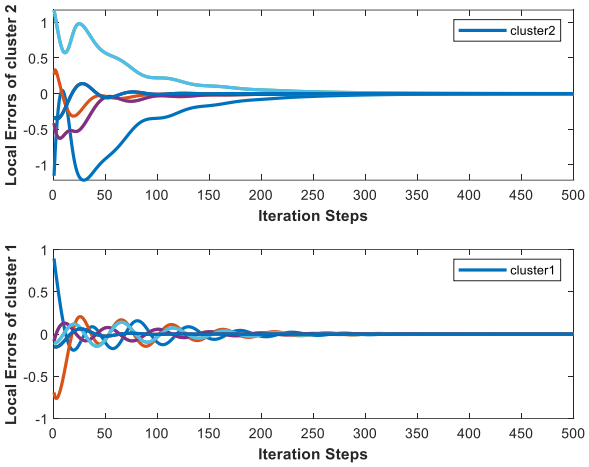

Fig. 6. Local errors of clusters 1 and 2

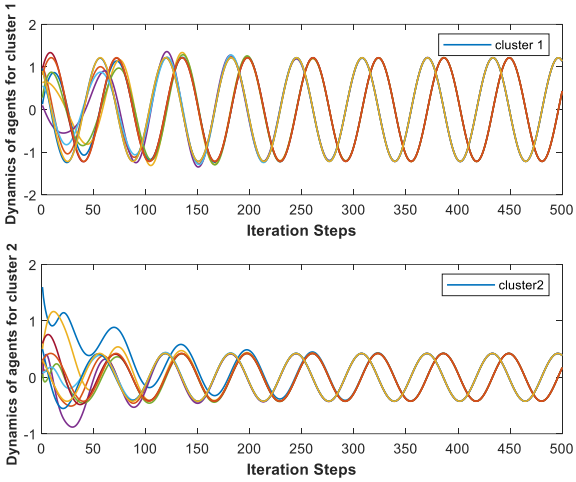

Fig. 7. Synchronization of follower agents to leader agents

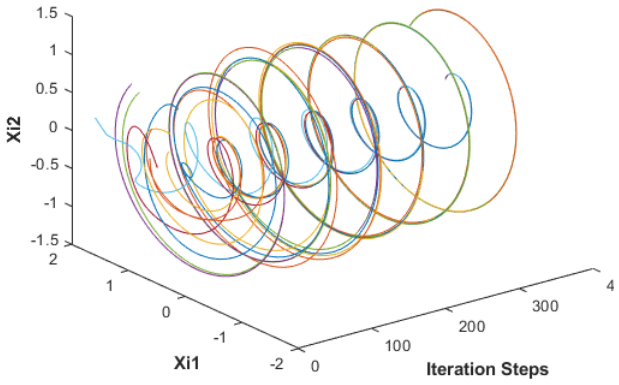

Fig. 8. The agents' and the virtual leaders' three-dimensional trajectory

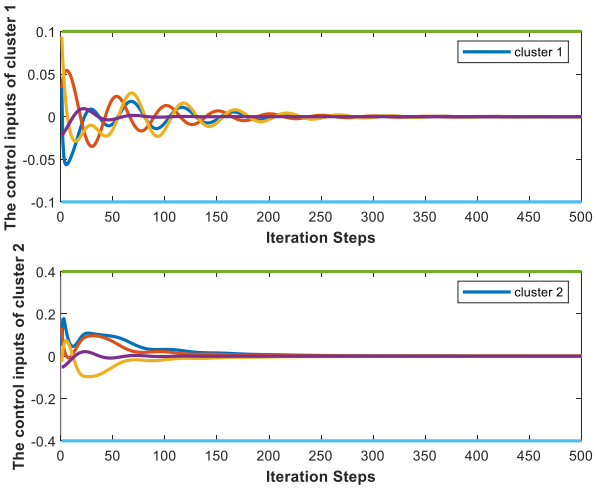

Fig. 9. Control inputs of clusters 1 and 2

Fig.s 2, 3, 4, and 5 illustrate the convergence of the control actor, disturbance actor, critic, and identifier for agent 1 in the first cluster and agent 5 in the second cluster, respectively. As shown in these figures, the weights of the critic-actor neural network have converged in both clusters. Fig. 6 displays the local errors for agents in the first and second clusters, all of which have converged to zero. The estimated states of all agents are presented in Fig. 7, demonstrating the synchronization of the states of all agents in a cluster with the leader of the same cluster while maintaining optimality. Fig. 8 depicts the three-dimensional trajectory of the agents and their virtual leaders. Finally, Fig. 9 shows the bounded control inputs for all agents in the first and second clusters. The results indicate that the proposed algorithm for solving cluster synchronization in discrete-time zero-sum graphical games with unknown constrained-input systems has successfully converged to approximate optimal solutions.

VIII. Results Analysis

This paper presents an innovative algorithm to address the cluster synchronization problem, allowing agents within each cluster to converge to their respective leaders. The presence of disturbances in the agents' dynamics adds realism to the proposed solution. Additionally, the issue of control input limitations is addressed, ensuring that the obtained inputs remain within an acceptable range. Remarkably, the proposed algorithm accomplishes all of this without requiring knowledge of the system dynamics.

In the following table, several references are compared with the present study.

TABLE I
Comparison of References with the Proposed Method

References	Graphical games	Zero-sum games	Cluster synchronization	Constrained-input systems
[20]	-	✓	-	-
[21]	✓	-	-	-
[28]	✓	-	✓	-
[29]	✓	-	✓	✓
proposed method	✓	✓	✓	✓

IX. Conclusion and Future Work

In this study, the synchronization problem of agents within each cluster for discrete-time zero-sum graphical games with unknown constrained input systems and external disturbances is addressed. An algorithm is presented that solves this problem without requiring knowledge of the system dynamics. To determine each agent's unknown dynamics, a neural network (NN) identifier is employed. Additionally, constraints on the control inputs are considered in the design method. The suggested approach is implemented as actor-critic structures to approximate the optimal value function, optimal control, and worst-case disturbance policies for the agents in each cluster. Simulation results demonstrate the efficacy of the proposed algorithm in synchronizing with the leader in each cluster while ensuring optimality.

In practical applications, MAS may encounter various challenges, including communication delays, packet loss, and system faults. Additionally, time delays between different groups of agents can significantly impact system performance. For example, in scenarios involving migrating geese or locust populations, agents within a group may arrive at a destination almost simultaneously, while agents from different groups may arrive at different times. This phenomenon is also evident in traffic management, where maintaining appropriate time delays between vehicles can help prevent congestion and accidents.

We have not addressed this issue in the current study, but it can be considered for future work. Furthermore, exploring event-triggered methods instead of traditional approaches and applying clustering methods to classify agents into groups are additional topics that warrant investigation in future research.

References

[1] T. Liu and Z. P. Jiang, “Distributed control of multi-agent systems with pulse-width-modulated controllers,” *Automatica*, vol. 119, pp. 109020–109020, Sep. 2020.

[2] X. Ge, Q. L. Han, L. Ding, Y. L Wang, X. M. Zhang, “Dynamic event-triggered distributed coordination control and its applications: A survey of trends and techniques,” *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 50, no. 9, pp. 3112-3125, 2020.

[3] P. Zhang, H. Xue, S. Gao, J. Zhang, “Distributed adaptive consensus tracking control for multi-agent system with communication constraints,” *IEEE Transactions on Parallel and Distributed Systems* vol. 32, no. 6 pp. 1293-1306, 2020.

[4] J. Huang, W. Wang, C. Wen, J. Zhou, G. Li, “Distributed adaptive leader-follower and leaderless consensus control of a class of strict-feedback nonlinear systems: A unified approach,” *Automatica*, vol. 118, pp. 109021, 2020.

[5] S. Zhai, X. Wang, W. X. Zheng, "Leaderless cluster consensus of second-order general nonlinear multiagent systems under directed topology." *IEEE Transactions on Circuits and Systems I: Regular Papers*, 2023.

[6] K. Chen, J. Wang, X. Zeng, Y. Zhang, F.L. Lewis, “Cluster output regulation of heterogeneous multi-agent systems,” *International Journal of Control*, vol. 93, no. 12, pp. 2973-2981, 2020.

[7] J. Qin, W. Fu, Y. Shi, H. Gao, Y. Kang, “Leader-following practical cluster synchronization for networks of generic linear systems: An event-based approach,” *IEEE Transactions on Neural Networks and Learning Systems*, vol. 30, no. 1, pp. 215-224, 2018.

[8] Y. Wang, Z. Ma, J. Cao, A. Alsaedi, F. E. Alsaadi, “Adaptive cluster synchronization in directed networks with nonidentical nonlinear dynamics,” *Complexity*, vol. 21, no. s2, pp. 380–387, 2016.

[9] Y. Han, W. Lu, T. Chen, “Cluster consensus in discrete-time networks of multi-agents with inter-cluster nonidentical inputs,” *IEEE Transactions on Neural Networks and Learning Systems* vol. 24, no. 4 pp. 566-578, 2023.

[10] Qin, Jiahu, and Changbin Yu. “Cluster consensus control of generic linear multi-agent systems under directed topology with acyclic partition,” *Automatica*, vol. 49, no. 9, pp. 2898-2905, 2013.

[11] T. Basar and Geert Jan Olsder, "Dynamic noncooperative game theory. 2nd ed. (Classics in applied mathematics 23)," Jan. 1999.
[12] K. G. Vamvoudakis, F. L. Lewis, and G. Hudas, "Multi-agent differential graphical games: Online adaptive learning solution for synchronization with optimality," *Automatica*, vol. 48, no. 8, pp. 1598–1611, Aug. 2012.
[13] Q. Wei, D. Liu, and F. L. Lewis, "Optimal distributed synchronization control for continuous-time heterogeneous multi-agent differential graphical games," *Information Sciences*, vol. 317, pp. 96–113, Oct. 2015.
[14] F. Tatari, M.B. Naghibi-Sistani, and K. G. Vamvoudakis, "Distributed optimal synchronization control of linear networked systems under unknown dynamics," *American Control Conference(ACC)*, May 2017.
[15] F. Tatari, M.B. Naghibi-Sistani, and K. G. Vamvoudakis, "Distributed learning algorithm for non-linear differential graphical games," *Transactions of the Institute of Measurement and Control*, vol. 39, no. 2, pp. 173–182, Jul. 2016.
[16] Q. Jiao, H. Modares, S. Xu, F.L. Lewis, and K.G. Vamvoudakis, "Multi-agent zero-sum differential graphical games for disturbance rejection in distributed control," *Automatica*, vol. 69, pp. 24-34, 2016.
[17] F. Tatari, K.G. Vamvoudakis, and M. Mazouchi, "Optimal distributed learning for disturbance rejection in networked nonlinear games under unknown dynamics," *IET Control Theory and Applications*, vol. 13, no. 17, pp.2838-2848, 2019.
[18] D. Liu, H. Li, and D. Wang, "Neural-network-based zero-sum game for discrete-time nonlinear systems via iterative adaptive dynamic programming algorithm," *Neurocomputing*, vol. 110, pp. 92–100, 2013.
[19] B. Kiumarsi, H. Modares, F.L. Lewis, and Z.P. Jiang, "H-infinity optimal control of unknown linear discrete-time systems: An off-policy reinforcement learning approach," *2015 IEEE 7th International Conference on Cybernetics and Intelligent Systems (CIS) and IEEE Conference on Robotics, Automation and Mechatronics (RAM)*, pp. 41-46, 2015.
[20] R. Song and L. Zhu, "Stable value iteration for two-player zero-sum game of discrete-time nonlinear systems based on adaptive dynamic programming," *Neurocomputing*, vol. 340, pp. 180–195, May 2019.
[21] M. Abouheaf, F. L. Lewis, K. G. Vamvoudakis, Sofie Haesaert, and R. Babuska, "Multi-agent discrete-time graphical games and reinforcement learning solutions," *Automatica*, vol. 50, no. 12, pp. 3038–3053, 2014.
[22] M. I. Abouheaf, F. L. Lewis, M. S. Mahmoud, and D. G. Mikulski, "Discrete-time dynamic graphical games: model-free reinforcement learning solution," *Control Theory and Technology*, vol. 13, no. 1, pp. 55–69, Feb. 2015.
[23] R. S. Sutton and A. G. Barto, "Reinforcement Learning: An Introduction," Robotica, vol. 17, no. 2, pp. 229-235, 1999.
[24] B. Kiumarsi, H. Modares, and F. Lewis, "Reinforcement Learning for Distributed Control and Multi-player Games," *In Handbook of Reinforcement Learning and Control*, *Springer*, Cham, pp. 7-27, 2021.
[25] H. Jiang, H. Zhang, G. Xiao, and X. Cui, "Data-based approximate optimal control for nonzero-sum games of multi-player systems using adaptive dynamic programming" *Neurocomputing*, 275, pp. 192-199, 2018.
[26] P. Liu, H. Zhang, C. Liu, and H. Su, "Online Dual-Network-Based Adaptive Dynamic Programming for Solving Partially Unknown Multi-Player Non-Zero-Sum Games With Control Constraints," *IEEE Access*, vol. 8, pp. 182295–182306, Jan. 2020.
[27] B. Luo, D. Liu, and H.-N. Wu, "Adaptive Constrained Optimal Control Design for Data-Based Nonlinear Discrete-Time Systems With Critic-Only Structure," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 29, no. 6, pp. 2099–2111, 2018.
[28] Yang, Ni, Jiang-Wen Xiao, Li Xiao, and Yan-Wu Wang. "Non-zero sum differential graphical game: cluster synchronization for multi-agents with partially unknown dynamics," *International Journal of Control*. vol, 92, no. 10 pp. 2408-2419, 2019.
[29] Z. Jahan, A. Dideban, M. Arab Khabori, " Cluster synchronization for unknown constrained-input discrete-time graphical games with reinforcement learning algorithms", *The first artificial intelligence and intelligent processing conference, Iran, Semnan*, 2022.
[30] S. Khoo, L. Xie, and Z. Man, "Robust Finite-Time Consensus Tracking Algorithm for Multirobot Systems," *IEEE/ASME Transactions on mechatronics*, vol. 14, no. 2, pp. 219–228, Mar. 2009.
[31] H. Zhang, Y. Luo, & D. Liu, "Neural-network-based near-optimal control for a class of discrete-time affine nonlinear systems with control constraints*"* *IEEE Transactions on Neural Networks*, vol. 20, no. 9, pp. 1490–1503, 2009.

APPENDIX

In this part, the flowchart of Algorithm 2 is presented.

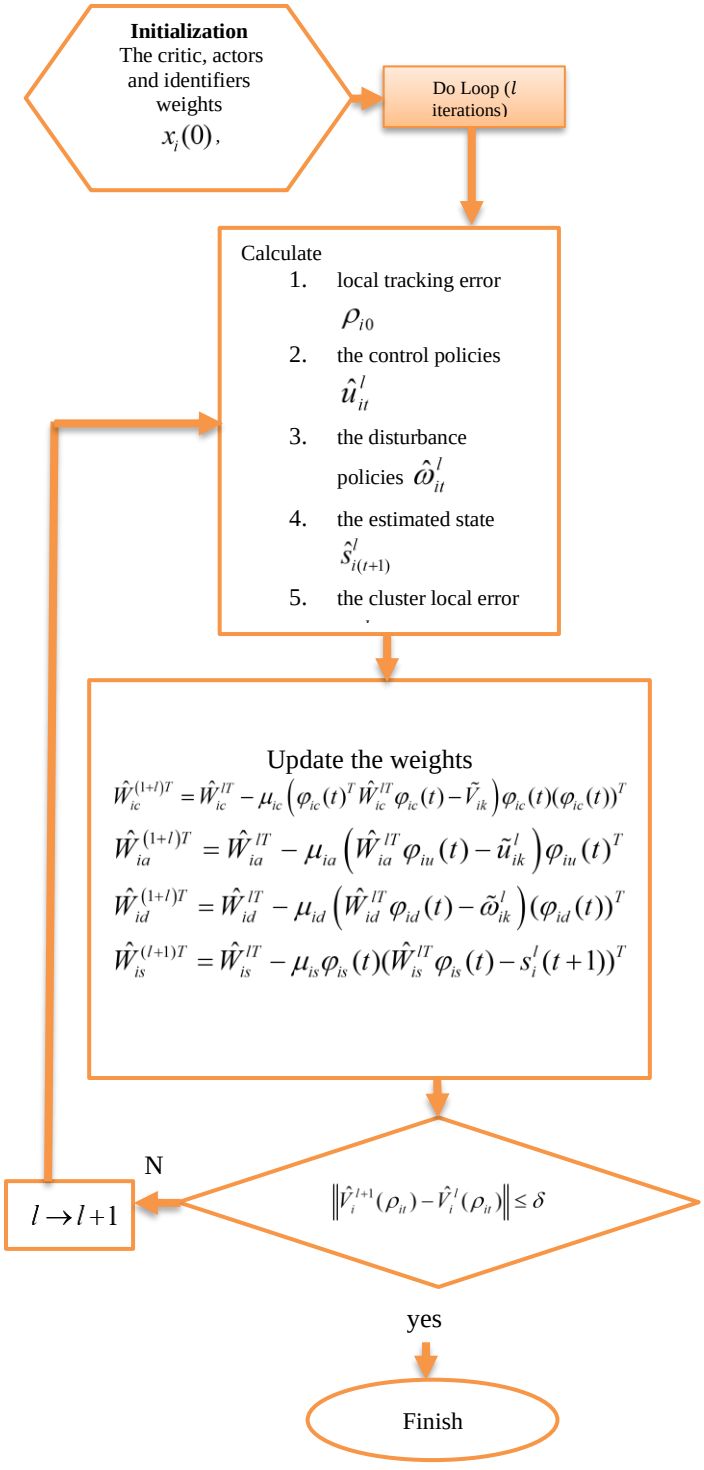

Fig. 10. Flowchart of Algorithm 2