2022

Autonomous Teaming of Multileader System - Robust Cluster Formation Approach

Maryam N. Naleini

University of Texas at Arlington

Department of Electrical Engineering

Doctorate Dissertation

Autonomous Teaming of Multileader System - Robust Cluster Formation Approach

Author:

Maryam Naleini

Doctoral advisor(s):

Frank Lewis, Professor, NAI

Ahmet Koru, Adjunct Professor

Graduation Semester: December 2022

Copyright Disclaimer and Letter of Intent

The purpose of this presentation is to clearly articulate my research aims and objectives and justify each choice in a concise manner. It will be also shown how this research outcomes will contribute to the current knowledge of Autonomous Systems & Communication Networks society.

Dedication

To my father,

Masoud Naleini,

06/30/1957 – 10/10/2022

who gave me the greatest gift of my life: He believed in me.

And to my husband,

Pouya,

who supported me and encouraged me to always do the best in life. He has given me the motivation and courage to be resilient.

Table of Contents

1 Abstract vi

2 Introduction 1

2.1 Optimal H∞ Literature Survey 1

2.2 Cluster Consensus Survey 2

2.3 Game Theory Algorithm 3

2.4 Integral Reinforcement Learning (IRL) 4

2.5 Contribution 4

2.6 Structure 6

3 Preliminaries 7

3.1 Graph Topology 7

3.2 Multi-agent System Dynamics 8

4 H∞ Optimization of Multileader multi-agent systems 10

4.1 Multiagent H∞ Optimal Performance 10

4.2 Disturbance Attenuation and Solutions for the H∞ Optimization Problem 11

5 Stability of Multileader Multi-agent Systems Cluster Consensus 13

5.1 Cluster Stability Analysis by Small Gain Theorem 13

5.2 Cluster Stability Analysis by Small Gain Theorem 14

6 Cluster Partitioning for Multileader Multi-agent Systems Cluster Consensus 19

6.1 Multileader Multi-agent System Cluster Consensus 19

6.2 Cluster Consensus of Multileader Multi-Agent Systems using H∞ optimal control policy 21

7 Integral Reinforcement Learning 23

7.1 Integral Reinforcement Learning for Optimal Adaptive Control of Continuous-Time Systems using Policy Iteration 23
7.2 Online Algorithm on an Actor-Critic-Disturbance Structure 25
7.3 Off-Policy IRL for Learning ML-MAS H∞ optimal problem 26
7.3.1 Off-Policy Reinforcement Learning Algorithm 26
7.3.2 Optimal Clustering Consensus Problem 27
7.4 ML-MAS Off-Policy IRL using Neural Networks 33
8 Simulation 37
8.1 Cluster Partitioning of Multileader Multi-agent Systems with (9) agents and (3) leaders 37
8.2 Cluster Partitioning of Multileader Multi-agent Systems with UUB Stability and Cluster Partitioning 45
8.3 ML-MAS Off-Policy IRL using Neural Networks 51
9 Conclusion 53
10 Publications 55
11 References and Footnotes 56
11.1 References 56
11.2 Footnotes 62

Table of Figures

Figure 1- Agent *i* inter-cluster and intra-cluster links 20
Figure 2- ML-MAS Graph Topology 38
Figure 3 - ML-MAS Synchronization – No UUB - Output trajectories of agents in each cluster - All Agents Synchronization - No cluster partitioning 41
Figure 4 – ML-MAS Synchronization – No UUB - Output trajectories of agents in each cluster - Cluster 1 Synchronization - No cluster partitioning 42
Figure 5 – ML-MAS Synchronization – No UUB - Output trajectories of agents in each cluster - Cluster 2 Synchronization - No cluster partitioning 43
Figure 6 – ML-MAS Synchronization – No UUB - Output trajectories of agents in each cluster - Cluster 3 Synchronization - No cluster partitioning 44
Figure 7 – ML-MAS Synchronization –UUB + Small Gain - Output trajectories of agents in each cluster – All Agents Consensus 46
Figure 8 – ML-MAS Synchronization –UUB + Small Gain - Output trajectories of agents in each cluster – Cluster 1 Consensus 47
Figure 9 – ML-MAS Synchronization –UUB + Small Gain - Output trajectories of agents in each cluster – Cluster 2 Consensus 48
Figure 10 – ML-MAS Synchronization –UUB + Small Gain - Output trajectories of agents in each cluster – Cluster 3 Consensus 49
Figure 11 – ML-MAS Cluster Consensus – Convergence of the control gain to its optimal value 51
Figure 12 – ML-MAS Cluster Consensus – Convergence of the kernel matrix *P* to its optimal value 52

Autonomous Teaming of Multileader System-Robust Cluster Formation Approach

1 ABSTRACT

The cluster consensus problem of Multileader MAS Is considered and the clustering problem of interconnected multileader systems is formulated as a disturbance attenuation problem. For the first time, it's proved that the multileader MAS can reach cluster consensus without limiting the communication between the clusters. The combination of Small Gain Theorem and H∞ optimization has been designed in the graphical differential game platform to prove the stability for the system. At the end, an online off-policy reinforcement learning algorithm is developed to find the solution to the $H\infty$ optimal problem of multileader MAS with completely unknown systems.

2 INTRODUCTION

The` standard consensus problem has been extensively studied in finding control laws that enable all agents to work cooperatively to reach stability [1]- [2]. Many of today's optimization problems in data science (including statistics, machine learning, and data mining) use distributed computation. Modern applications have access to real-life applications which cannot be handled by a single processor alone. This will encourage us to combine the knowledge from computer science, behavioral science and finance and distributing data among multiple agents and processing it in a decentralized manner based on the available local information. The applications in statistical learning along with other applications in distributed data processing where information is inherently distributed among many processors (i.e. distributed sensor networks, coordination, and cooperation distribution control [3]) have been studied vigorously on distributed multiagent optimization.

2.1 OPTIMAL H∞ LITERATURE SURVEY

The H∞ optimal control policies have been defined to attenuate the effect of disturbances on the performance function. The focus of H∞ control theory has been established on designing regulators to drive the states of the system to zero in the presence of disturbance [4]- [5].

In practice, however, it is often required to force the states or outputs of the system to track a reference trajectory. Existing solutions to the H∞ tracking problem are composed of two steps. First, a feedforward control input is designed to guarantee the perfect tracking. Second, a feedback control input is designed by solving a Hamilton–Jacobi–Isaacs (HJI) equation to stabilize the tracking error dynamics. These methods are suboptimal as they ignore the cost of the feedforward control input in the performance function. Moreover, in these methods, procedures for computing the feedback and feedforward terms are based on the offline solution methods that require complete knowledge of the system dynamics [6]- [7].

2.2 CLUSTER CONSENSUS SURVEY

Cluster analysis have been widely studied in various fields. Papadimitriou studied computing equilibria in multiplayer games [8]. Olfati-Saber presented a theoretical framework for design and analysis of distributed flocking algorithms [9]. Delgado and Stern have studied geographical clustering for economic studies in different industries [10]. Hansen has done a survey of clustering analysis from a mathematical programming viewpoint [11].

Clustering is considered as the first step in data analysis in computer science. Many different clustering methods has been developed [12]- [13] such as hierarchical agglomerative clustering, mixture densities, graph partitioning, and spectral clustering. Most clustering methods focus on finding a single optimal or near-optimal clustering according to some specific clustering criterion. Nguyen and Caruana addressed the problem of combining multiple clustering without access to the underlying features of the data called clustering consensus [14].

The multiagent system consensus has been studied through multiple approaches. Movric and Lewis designed a distributed cooperative control protocol reach consensus in the multiagent system [15]. Han studied cluster consensus in continuous-time networks of multi-agents with time-varying topologies via non-identical inter-cluster inputs [16]. Qin and Yu investigated the cluster consensus control for generic linear multi-agent systems under directed interaction topology with acyclic partition via distributed feedback controller [17].

The consensus problems have been usually addressed by eliminating the inter-cluster couplings or using the zero-row sum assumption to ensure the stability [18]. K. Chen Compared the standard consensus to the cluster consensus featured with the inter-cluster couplings.

Once the inter-cluster couplings are eliminated, the clustering problem is reduced to multiple standard consensus problems. in this paper, the goal is to determine the conditions in the dynamics of the agents and the communication graph that allows the agents to synchronize with their group without removing the inter-cluster communications.

The cluster consensus was studied for the strong intra-cluster couplings and the lower bounds for the coupling strengths within clusters has been defined to ensure cluster synchronization [19]. The results of [19] are invalid when the couplings within each cluster are too weak. In a later year, Qin and Yu proved that if clusters interact with each other in an acyclic mode, the strengths of couplings within the same cluster will not affect the cluster consensus behavior. They have relaxed

the acyclic mode assumption by introducing intra-cluster balanced topologies with antisymmetric. However, satisfying the acyclic partition assumptions on the topology limits the communication structures.

2.3 GAME THEORY ALGORITHM

The interaction of complex interconnected systems can be studied using the mathematical framework of game theory [20]- [21]. The agents involved can have cooperative and conflicting objectives, and their decisions are based upon optimizing individual payoffs functions. Graphical games [22] model the interactions among players that communicate using a graph network topology. Multiplayer games approaches have been used, for example, to optimally allocate the resources of optical networks [23], design optimal motion planning for multiple robots with different goals [24], and to protect a network from adversaries [25].

Strategies for team decision problems, including N-player games, are normally solved offline by solving the coupled Hamilton–Jacobi (HJ) equations for nonlinear systems or coupled Riccati equations for linear systems. These procedures usually require complete knowledge of the system dynamics, and the computational burden grows exponentially with the state-space dimension. Moreover, using offline approaches prevent the players from being able to change their objectives in real time [26].

Given the nature of the interactions and the fact that the environment is highly uncertain and dynamic, enabling autonomous agents to gracefully adapt their decision-making strategies is of paramount importance. Reinforcement Learning (RL) [27] is learning technique that does not require a model of the agents or the environment and can be used online in real time. These characteristics make RL well suited for multiplayer games, where each agent knows little about other agents in the game. Using RL, the performance of an individual gradually improves as it learns from the observed responses of its behavior in its environment [28]. In this article the multiagent system has been modeled as multiplayer games and the games have been solved online by adaptive learning in real time using data measured along the trajectories of the agents. The full dynamics of the agents do not need to be known for these online solution techniques. Game-based architecture approach implicitly solve the required game architecture equations without ever explicitly solving them.

2.4 INTEGRAL REINFORCEMENT LEARNING (IRL)

Studying uncertain dynamical systems is not only practical but a means of addressing the control problem for a large class of nonlinear systems based on a simplified model [29]. In [30] design tools have been introduced which allow us to address the problem of stabilizing systems with intricate structure. it has been proven that the uncertain dynamical system can be robustly stabilized by means of partial-state feedback.

Modares and Lewis [31] describes the use of principles of reinforcement learning to design feedback controllers for discrete- and continuous-time dynamical systems that combine features of adaptive control and optimal control. Adaptive control and optimal control represent different philosophies for designing feedback controllers. Optimal controllers are normally designed offline by solving Hamilton–Jacobi–Bellman (HJB) equations. Determining optimal control policies for nonlinear systems requires the offline solution of nonlinear HJB equations, which are often difficult or impossible to solve. By contrast, adaptive controllers learn online to control unknown systems using data measured in real time along the system trajectories. Adaptive controllers are not usually designed to be optimal in the sense of minimizing user-prescribed performance functions. Indirect adaptive controllers use system identification techniques to first identify the system parameters and then use the obtained model to solve optimal design equations. Adaptive controllers may satisfy certain inverse optimality conditions.

2.5 CONTRIBUTION

In this dissertation, the cluster consensus problem of Multileader has been considered. Chen has studied the heterogeneous MASs considering the heterogeneous dynamics and the negative couplings among agents [32]. Chen has restricted the communication topology between the cluster using zero-row sum assumption for Laplacian matrix to assure the cluster has no other Impact on other clusters. He designed the problem using Hamiltonian performance optimization. In this paper, the Chen's restrictions on the communication topology have been removed which allow the clusters to communicate with each other freely through the optimization. We have also used Min-Max differential game theory to optimize the state feedback control system. An important idea in

this paper is to formulate the clustering problem of interconnected multileader systems as a disturbance attenuation problem as formulated in [5].

There have been several studies in the past to reach consensus in multiagent system. In this paper, for the first time, it has been proved that the multileader MAS can reach cluster consensus without limiting the communication between the clusters. The combination of Small Gain Theorem and H∞ optimization has been designed in the graphical differential game platform to prove the stability for the system.

After proving the cluster consensus for Multileader MASs, an online off-policy reinforcement learning algorithm is developed to find the solution to the $H\infty$ optimal problem of multileader MAS with completely unknown systems. The leaders and the agents of other clusters outside each agent cluster will be defined as the system disturbance. It is not required that the disturbance be adjustable. An augmented system is constructed from the tracking error dynamics and the

command generator dynamics for the H∞ optimal performance problem.

A performance HJI equation associated with the discounted performance function is derived, which gives both the feedforward and feedback parts of the control input simultaneously. An upper-bound and lower-bound is obtained for the discount factor to assure local asymptotic stability of the error dynamics using **Ultimately Uniformly Bounded (UUB)**. An off-policy RL algorithm is then developed to find the solution to the HJI equation online using only the measured data and without any knowledge about the system dynamics. Convergence of this algorithm to the solution to the HJI equation is shown.

The major contributions of this paper are as follows:

Multileader MASs cluster consensus has been proved under general topology with existing spanning tree, with NO limiting assumption on the communication graph. The underlying mechanisms between the system dynamics and the communication graph to reach cluster consensus are presented.

The cluster consensus problem of Multileader MASs is investigated, where all clusters are allowed to have different communication weights. This extends existing results in [32] for clustering consensus.

an online off-policy reinforcement learning algorithm is developed to find the solution to the $H\infty$ optimal problem of multileader MAS with completely unknown systems. This will extend the off-policy RL in [30] to Multileader MAS.

2.6 STRUCTURE

This article has been structured as follows:

1. Preliminaries on graph topology and the system dynamics and all the notations are provided in Section II.
2. The Multileader MAS optimal problem and the state feedback control protocol are defined Sections III.
3. The Multileader MAS cluster consensus has been formulated in section IV using small gain theorem and Ultimately Uniformly Bounded (UUB) stability solution.
4. An online off-policy reinforcement learning algorithm is developed to find the solution to the H∞ optimal problem of multileader MAS with completely unknown systems in section V.
5. The proposed cluster consensus method and off-policy IRL method are both applied to a linear system to show that it converges to the optimal solution in section VI.

3 PRELIMINARIES

In this section we present the definitions and the mathematical background used throughout the paper.

3.1 GRAPH TOPOLOGY

A graph is a pair $G=(V,E)$ with $V=\{v_1,...,v_N\}$ a set of N nodes or vertices and E a set of edges or arcs. The elements of E are denoted as (v_i,v_j) which represents an edge or arc from v_i to v_j and is depicted as an arrow with tail at v_i and head at v_j. The edges represent the allowed flow of information in the graph. We assume the graph is simple, i.e. $(v_i,v_j)\notin E,\forall i$ (no self-loops), and no multiple edges between the same pairs of nodes. The set of neighbors of a node v_i is $N_i=\{v_j:(v_i,v_j)\in E\}$

Let the graph G be partitioned in P disjoint clusters. The cluster to which agent i belongs is denoted as C_i, and C_{-i} is the set of all other clusters that agent i does not belong to. The notation $j\in C_i$ means that agent j belongs to the same cluster as i, and $k\in C_{-i}$ accounts for the agents k that do not belong to the same cluster as agent i. Agents from different clusters can be neighbors of each other in the graph topology G.

Associated with each edge $(v_i,v_j)\notin E,\forall i,j$ is a weight $e_{ij}\geq 0$ which represents the weight of the link from agent i to agent j. Let $e_{ij}>0$ only if there is an edge from node j to node i, and $e_{ij}=0$ otherwise. The adjacency matrix is defined as $\mathcal{A}=(e_{ij})_{N\times N}$. Represent the weight e_{ij} as $a_{ij}=e_{ij}$ when agent i and agent j belong to the same cluster and as $b_{ik}=e_{ij}$ when agent i and agent k belong to different clusters.

The in-degree matrix $D\in\mathbb{R}^{N\times N}$ is a diagonal matrix with the i^{th} diagonal element being the in-degree of node i, defined as $d_i=\sum_{j\in N_i}e_{ij}$. The Laplacian matrix of G is defined as $L=D-\mathcal{A}$.

3.2 MULTI-AGENT SYSTEM DYNAMICS

Consider a system consisting of N agents with homogeneous linear dynamics as

$$\dot{x}_i = Ax_i + Bu_i, i = 1,..., N \tag{1}$$

and a set of P leaders, regarded as the leader nodes, with dynamics

$$\dot{x}_p = Ax_p, p \in P \tag{2}$$

where $A \in \mathbb{R}^{n\times n}, B \in \mathbb{R}^{n\times m}, x_i \in \mathbb{R}^n$ is the state of agent i, and $u_i \in \mathbb{R}^r$ is the control input for agent i.

Each agent i is represented by a node in the graph G defined in the previous subsection. Each leader state, x_p, provides the desired dynamics for the agents in cluster p, where $p \in P$ set of the leaders. The cluster consensus problem has been described for every agent in cluster p as achieving synchronization with the corresponding leader state x_p.

In this document, when talking about specific agent i. we reserve the sub-index j and p for agent and leader inside the same cluster as i respectively, and sun-indices k and l for the agents and leaders outside the cluster of agent i.

The local synchronization error δ_i of agent i is defined based on the defined communication of agent i with other agents and the leaders

$$\delta_i = \sum_{j\in C_i}^{N} a_{ij}(x_j - x_i) + \sum_{\substack{p\in C_i \\ p\in P}}^{P} g_{ip}\left(x_p - x_i\right) + \sum_{k\in C_{-i}} b_{ik}\left(x_k - x_i\right) + \sum_{\substack{l\in C_{-i} \\ l\in P}} g_{il}\left(x_l - x_i\right) \tag{3}$$

where a_{ij} and b_{ik} are the communication link weights between the agents and g_{ip} and g_{il} are the pinning gain weights from the leaders.

The local error δ_i has been expressed to represent the cluster partition of the multiagent system. It is formulated in four terms: (1) The communication between agent i and the agents in the same cluster as i , (2) the intra-cluster link between agent i and the cluster i leader, (3) the inter-cluster set of links between the agent i and other clusters' leaders, and (4) the inter-cluster links between agent i and the agents outside its cluster.

The dynamics of the local error in equation (3) is given by

$$\dot{\delta}_i = \sum_{\substack{j\in C_i \\ j=1}}^{N} a_{ij}(\dot{x}_j - \dot{x}_i) + \sum_{\substack{p\in C_i \\ p\in P}}^{P} g_{ip}\left(\dot{x}_p - \dot{x}_i\right) + \sum_{\substack{l\in C_{-i} \\ l\in P}} g_{il}\left(\dot{x}_l - \dot{x}_i\right) + \sum_{k\in C_{-i}} b_{ik}\left(\dot{x}_k - \dot{x}_i\right) \tag{4}$$

By replacing the system dynamics (1), (2) and operating algebraically, the local error dynamics can be written as

$$\dot{\delta}_i = A\delta_i - \left(d_i + g_i\right) Bu_i + \sum_{j \in C_i} a_{ij} Bu_j + \sum_{k \in C_{-i}} b_{ik} Bu_k \tag{5}$$

4 H∞ Optimization of Multileader multi-agent systems

In this section, we present an H∞ optimal design for a multi-leader homogeneous MAS using state variable feedback control. We then determine the Hamilton-Jacobi-Issacs (HJI) equations that provide the optimal control policies for the agents.

The goal of H∞ optimization is to attenuate the effect of all agents and the leaders on the performance of an agent and optimize its performance.

4.1 Multiagent H∞ Optimal Performance

The H∞ control problem can be formulated as a zero-sum differential game [33]. The optimal control policy solutions, determined in the following subsection, provide the saddle point solution to the differential game [34].

Definition 1- *(Bounded L2-Gain)* Consider the system with output *y(x(t))* and a performance output *z(t)*. In the bounded L2-gain problem [33], one desires to find a feedback control policy u(x) such that, when x(0) =0 and for all disturbances $d(t) \in L2[0,\infty)$ one has

$$\frac{\int_t^\infty \|z(\tau)\|^2 \, d\tau}{\int_t^\infty \|d(\tau)\|^2 \, d\tau} \le \gamma^2 \tag{6}$$

for a prescribed $\gamma > 0$ and for all $T > 0$. That is, the L2-gain from the disturbance to the performance output is less than or equal to γ. The H∞ control problem is to find, if it exists, the smallest value $\gamma^* > 0$ such that for any $\gamma > \gamma^*$, the bounded L2-gain problem has a solution. In the linear case an explicit expression can be provided for the H∞ gain [35]. To solve the bounded L2-gain problem, the zero-sum game just developed. In this zero-sum game, both inputs can be controlled, with the control input seeking to minimize a performance index and the disturbance input seeking to maximize it. By contrast, here *d(t)* is a disturbance that cannot be controlled, and *u(t)* is the control input used to offset the deleterious effects of the disturbance.

Before defining the H∞ optimization problem, we define the consensus condition.

Definition 2- *(Cluster Consensus Protocol)* The H∞ optimal problem is to design local control protocol u_i, u_j, u_k for all the agents in each cluster, such that all the agents in each cluster reach consensus with their leader.

The main difference between definition 2 and the standard definition of consensus is that the problem here is defined for multi-leader condition.

Definition 3- *(Graphical H∞ Optimal Policy)* The optimal policy in graphical game defines how an agent prepares its best response if its neighbors will attempt to maximize its performance index. As this is usually not the strategy followed by such neighbors during the game, every agent can

expect to achieve a better performance than its minmax value. To determine the performance index, we formulate a zero-sum game between agent i and its neighbors inside and outside the cluster.

The performance index for each agent is defined as

$$J_i\left(\delta_i(0),u_i,u_{-i}\right)=\frac{1}{2}\int_0^\infty\left[\delta_i^T Q_i\delta_i+u_i^T R_i u_i\right]+\frac{1}{2}\int_0^\infty\left[-\gamma_1^2\sum_{j\in C_i}a_{ij}u_j^T R_j u_j-\gamma_2^2\sum_{k\in C_{-i}}b_{ik}u_k^T R_k u_k\right] \quad (7)$$

where $\gamma_1^2>0$ and $\gamma_2^2>0$ are predefined constant parameters.

Remark 1- Performance index (7) represents our new approach to solve the multi-leader consensus problem. Here, the behavior of the neighbors of the agent *i* are taken as a disturbance to the system. As in the H∞ approach, we determine the optimal control policy against the worst-case disturbances, i.e., the policies u_j, u_k that maximize (7).

Remark 2- The disturbance attenuation conditions $\gamma_1^2>0$ and $\gamma_2^2>0$ imply that the effect of the disturbance input to the desired performance output is attenuated by a degree at least equal to $\gamma_1^2>0$ for u_j and $\gamma_2^2>0$ for u_k. The minimum value of $\gamma_1^2>0$ and $\gamma_2^2>0$ for which the disturbance attenuation condition is satisfied give the optimal robust control solution [36].

4.2 Disturbance Attenuation and Solutions for the H∞ Optimization Problem

In this section, the Graphical Game Algebraic Riccati Equation (GARE) is formulated, which gives the solution to the H∞ optimization problem stated in section *A*.

The Bellman equation for the performance function (7) can be defined in terms of the Hamiltonian as follows

$$H_i=\frac{1}{2}\left[\delta_i Q_i\delta_i+u_i^T R_i u_i-\gamma_1^2\sum_{j\in C_i}a_{ij}u_j^T R_j u_j\right]+\frac{1}{2}\left[-\gamma_2^2\sum_{k\in C_{-i}}b_{ik}u_k^T R_k u_k+\nabla V_i^T\left(\delta_i\right)\dot{\delta}_i\right]=0 \quad (8)$$

Assuming that the value function has a quadratic form as $V_i\left(\delta_i\right)=\delta_i^T P_i\delta_i$, then $\nabla V_i\left(\delta_i\right)=2P_i\delta_i$ and the Hamiltonian can be written as

$$\begin{aligned}H_i=&\frac{1}{2}\left[\delta_i Q_i\delta_i+u_i^T R_i u_i\right]+\frac{1}{2}\left[-\gamma_1^2\sum_{j\in C_i}a_{ij}u_j^T R_j u_j-\gamma_2^2\sum_{k\in C_{-i}}b_{ik}u_k^T R_k u_k\right]\\&+\delta_i^T P_i\left[A-\left(d_i+g_i\right)Bu_i+\sum_{j\in C_i}a_{ij}Bu_j+\sum_{k\in C_{-i}}b_{ik}Bu_k\right]\end{aligned} \quad (9)$$

A necessary condition for optimality with respect to the control input u_i is the stationary condition $\frac{\partial H_i}{\partial u_i} = 0$. This procedure yields the optimal control policy for agent *i* as

$$u_i = \left(d_i + g_i\right) R_i^{-1} B^T P_i \delta_i \tag{10}$$

In turn, the worst-case disturbance of our system is given by the neighbor policies u_j and u_k that maximize the performance index (7), and that are obtained by $\frac{\partial H_i}{\partial u_j} = 0$ and $\frac{\partial H_i}{\partial u_k} = 0$ as

$$\upsilon_j = \frac{1}{\gamma_1^2} R_j^{-1} B^T P_i \delta_i \tag{11}$$

and

$$\upsilon_k = \frac{1}{\gamma_2^2} R_k^{-1} B^T P_i \delta_i \tag{12}$$

respectively. Notice that v_j and v_k are not the actual optimal policy for agent j and k, but only represent the worst-case policy for the performance function.

Substituting these control policies in the Bellman equation (8) we get the HJI equation

$$\begin{aligned} &\frac{1}{2}\delta_i^T \left[Q_i + \left(d_i + g_i\right)^2 P_i B R_i^{-1} B^T P_i - \frac{1}{\gamma_1^2} \sum_{j \in C_i} a_{ij} \left(P_i B R_j^{-1} B^T P_i \right) - \frac{1}{\gamma_2^2} \sum_{k \in C_{-i}} b_{ik} \left(P_i B R_k^{-1} B^T P_i \right) \right] \delta_i \\ &+ \delta_i^T \left[\frac{1}{2}\left(P_i A + A^T P_i \right) - \left(d_i + g_i\right)^2 P_i B R_i^{-1} B^T P_i + \frac{1}{\gamma_1^2} \sum_{j \in C_i} a_{ij} \left(P_i B R_j^{-1} B^T P_i \right) + \frac{1}{\gamma_2^2} \sum_{k \in C_{-i}} b_{ik} \left(P_i B R_k^{-1} B^T P_i \right) \right] \delta_i = 0 \end{aligned} \tag{13}$$

This implies that matrix P_i in the policies (10)–(12) solves the Game Algebraic Riccati equation (**GARE**)

$$Q_i - \left(d_i + g_i\right)^2 P_i B R_i^{-1} B^T P_i + \frac{1}{\gamma_1^2} \sum_{j \in C_i} a_{ij} P_i B R_j^{-1} B^T P_i + \frac{1}{\gamma_2^2} \sum_{k \in C_{-i}} b_{ik} P_i B R_k^{-1} B^T P_i + P_i A + A^T P_i = 0 \tag{14}$$

From the control policy (10) above, the control gain matrix for agent *i* can be expressed by defining

$$K_i = \left(d_i + g_i\right) R_i^{-1} B^T P_i \tag{15}$$

such that $u_i = K_i \delta_i$.

Remark 3- Graphical game H∞ distributed control policies for the agents if there exist positive definite solutions P_i for the equation (14) respectively.

In the following section, we analyze the stability properties of the closed-loop system using the designed H∞ policies for multileader clustering.

5 STABILITY OF MULTILEADER MULTI-AGENT SYSTEMS CLUSTER CONSENSUS

In this section, we first prove that the cluster consensus error dynamics (16) are ultimately uniformly bounded (UUB) [37] when we use the control policies (10). Then, a cluster partitioning technique is used to guarantee the asymptotic stability of the agents to synchronize with their corresponding leaders.

5.1 CLUSTER STABILITY ANALYSIS BY SMALL GAIN THEOREM

In this section, we first introduce the small gain theorem approach for MAS and then, by using the small gain theorem, and then, in section *B*, we prove ultimately uniformly bounded (UUB) stability.

Consider the local error dynamic defined in (17) and let all agent $i=1,\ldots,N$ use control policies of the form (10). For nominal matrices A_i^{CP} and *B*, the closed-loop system dynamic for each agent is

$$\dot{\delta}_i = A_i^{CP}\delta_i + B\varpi_i \tag{18}$$

where A_i^{CP} is the closed loop system matrix for the agent *i* defined as

$$A_i^{CP} = A - \left(d_i + g_i\right)BK_i \tag{19}$$

and the disturbance term is defined as

$$\varpi_i = \sum_{j\in C_i} a_{ij}K_j\delta_j + \sum_{k\in C_{-i}} b_{ik}K_k\delta_k \tag{20}$$

where matrices K_j and K_k are as in (15).

In (18), the influence of the neighbor control policies u_j and u_k is taken as a disturbance.

The small gain theorem [38] is employed to analyze the stability properties of the ML-MAS.

For Multi-leader MAS defined in (18), The transfer function for agent *i* is

$$T_i(s) = \left(sI - A_i^{CP}\right)^{-1}B \tag{21}$$

Taking other leaders and the agents from other clusters as the disturbance to be attenuated, the input–output relationships of the local system is represented by (21).

Define the matrix H_i

$$H_i(s)=\begin{bmatrix} a_{i1}K_1\delta_1 & a_{i2}K_2\delta_2 & \cdots & 0 & \cdots & a_{ip}K_p\delta_p \end{bmatrix} \tag{22}$$

where the i^{th} term of the matrix H_i is zero. Matrix H is defined for inter-cluster couplings as

$$H(s)=\begin{bmatrix} H_{i1} & \cdots & H_{ij} & \cdots & H_{pp} \end{bmatrix} \tag{23}$$

The H∞-norm of the transfer function matrix for system (18) is

$$\|T_i(s)\|_\infty = \sup_{\omega\in R} \sigma^{-} \quad [T_i(j\omega)] \tag{24}$$

The small gain theorem is used to solve the H∞ optimal problem for ML-MAS system.

Lemma 1- [39] *Small Gain Theorem for Multi-leader MAS:* Consider the ML-MAS defined in (1) and (2). For $\gamma^2 > 0$, the following are equivalent:

1) A_i^{CP} is Hurwitz and $\|T_i(s)\|_\infty < \gamma^2$

2) There exists a $P > 0$ such that

$$A^T P + PA + \frac{1}{\gamma^2} PBB^T P + I < 0 \tag{25}$$

The closed-loop system dynamic (18) is finite-gain L_2 stable if 1 or 2 holds.

Remark 5- The ML-MAS optimal problem is solved if A_i^{CP} is Hurwitz, and for $\rho(H_i) \neq 0$

$$\|T_i(s)\|_\infty < \frac{1}{\rho(H_i)} \tag{26}$$

This only depends on the inter-cluster couplings H_i. Therefore, the gain matrices shall be selected to assure the H∞ norm exists, and it is smaller than $\frac{1}{\rho(H_i)}$.

5.2 Cluster Stability Analysis by Small Gain Theorem

In this section, optimal cluster consensus of ML-MASs is discussed in the framework of multiagent graphical games. It is shown how to find optimal protocols for every agent. It is also shown that the optimal response makes all agents synchronize to the leader and reach an ultimately uniformly bounded (UUB) stability.

Assumption 1- The predefined constant for all clusters can be defined as the largest for all $\gamma_1^2 = \gamma_2^2 = \gamma^2 > \gamma^*$ Where γ^* is as definition 1.

Assumption 2- The performance index (7) is selected such that

$$Q_i \geq I \tag{27}$$

From Assumption 2, U_i is orthogonal and the H∞ norm of the transfer function matrix is less than $\gamma^2 > 0$ [40].

Theorem 1 shows asymptotic stability for all agents once they select their own optimal response to the cluster leaders.

Theorem 1 – ML-MAS Cluster Synchronization: Consider the ML-MAS with system dynamics given by (1) and (2).

$$\dot{x}_i = Ax_i + Bu_i, i = 1,\ldots,N$$
$$\dot{x}_p = Ax_p, p \in P$$

Assume the communication graph contains a spanning tree. The optimal control policy protocols $u_i = (d_i + g_i) R_i^{-1} B^T P_i \delta_i$ (10) make the local errors (3)

$$\delta_i = \sum_{j \in C_i}^{N} a_{ij}(x_j - x_i) + \sum_{\substack{p \in C_i \\ p \in P}}^{P} g_{ip}(x_p - x_i) + \sum_{k \in C_{-i}} b_{ik}(x_k - x_i) + \sum_{\substack{l \in C_{-i} \\ l \in P}} g_{il}(x_l - x_i)$$

Asymptotically stable for all agent *i*.

Proof: The stability of H∞ optimal solution is reached using small gain theorem in *lemma 1*.

To find the condition to assure the stability of the error dynamics, the small gain theorem must hold. Equation (14) can be written as

$$Q_i + P_i B\left[-(d_i + g_i)^2 R_i^{-1} + \frac{1}{\gamma_1^2}\sum_{j \in C_i} a_{ij} R_j^{-1} + \frac{1}{\gamma_2^2}\sum_{k \in C_{-i}} b_{ik} R_k^{-1}\right] B^T P_i + P_i A + A^T P_i = 0 \tag{28}$$

From assumption 2, we know $Q_i \geq I$. Using the optimal control policy for agent *i* (10), $K_i = (d_i + g_i) R_i^{-1} B^T P_i$, equation (28) will be redefined as

$$A^T P_i + P_i A - P_i B\left[-(d_i + g_i)^2 R_i^{-1} + \frac{1}{\gamma_1^2}\sum_{j \in C_i} a_{ij} R_j^{-1} + \frac{1}{\gamma_2^2}\sum_{k \in C_{-i}} b_{ik} R_k^{-1}\right] B^T P_i + I < 0 \tag{29}$$

Comparing GARE equation (14) and equation (29), and using assumption 2, For the closed loop system, equation (27) will hold if

$$-\left(d_i+g_i\right)^2 R_i^{-1}+\frac{1}{\gamma_1^2}\sum_{j\in C_i} a_{ij}R_j^{-1}+\frac{1}{\gamma_2^2}\sum_{k\in C_{-i}} b_{ik}R_k^{-1}>\frac{1}{\gamma^2}I \tag{30}$$

From assumption 1, If $\gamma^2=\gamma_1^2=\gamma_2^2$, equation (30) can be summarize as

$$-\frac{1}{\gamma^2}I+\frac{1}{\gamma^2}\sum_{j\in C_i} a_{ij}R_j^{-1}+\frac{1}{\gamma^2}\sum_{k\in C_{-i}} b_{ik}R_k^{-1}>\left(d_i+g_i\right)^2 R_i^{-1} \tag{31}$$

For the inequality (31) to hold, the following equation should hold for the bounded L_2-Gain value.

$$\gamma^2<\frac{\sum_{j\in C_i} a_{ij}R_j^{-1}+\sum_{k\in C_{-i}} b_{ik}R_k^{-1}-1}{\left(d_i+g_i\right)^2 R_i^{-1}} \tag{32}$$

Lemma 1 proves that for $\gamma^2>0$, the system is stable and $\|T_i(s)\|_\infty<\gamma^2$. Therefore, the small gain theorem holds and

$$0\le\gamma^2\le\frac{1}{\rho\left(H_i\right)} \tag{33}$$

where H_i is the matrix of inter-cluster links. While A is Hurwitz,

$$\left\|T_i(s)\right\|_\infty<\frac{1}{\rho\left(H_i(s)\right)} \tag{34}$$

Remark 6- While ML-MAS system is stable, there always exists a solution $P_i>0$ such that the GARE equation (14) holds.

We have proved the small gain theorem holds and there exists a bounded L_2-gain value. In theorem 2, it is proved the ML-MAS is Ultimately Uniformly Bounded for all agents.

Theorem 2- *UUB Stability of ML-MAS:* There exists a constant $\Upsilon>0$ and the time function T such that for all agent i in ML-MAS, the norm of the local error dynamic δ_i is uniformly bounded, and by applying control policies $u_i,\forall i\in N$ to the system, ML-MAS will reach UUB (Ultimately Uniformly Bounded) stability.

Proof: For all agent *i*, the local synchronization error δi of agent i is defined based on the defined communication of agent i with other agents and the leaders in equation (1) and (2). In order to reach stability, the local synchronization error δ_i must go to zero.

$$\delta_i \to 0 : \delta_i = \left\langle \sum_{j \in C_i}^{N} a_{ij}(x_j - x_i) + \sum_{\substack{p \in C_i \\ p \in P}}^{P} g_{ip}\left(x_p - x_i\right) + \sum_{k \in C_{-i}} b_{ik}\left(x_k - x_i\right) + \sum_{\substack{l \in C_{-i} \\ l \in P}} g_{il}\left(x_l - x_i\right) \right\rangle \to 0 \qquad (35)$$

Assume x_{P1} and x_{P2} are the state of the leaders of cluster *1* and *2*. It is the state of the leader for different cluster cannot be the same, the following statement is accurate.

$$x_{P1} \neq x_{P2} \qquad (36)$$

Since the state of agent *i* in cluster *P* is related to the leader in its own cluster and the neighboring clusters, the agent *i*'s state x_i will not reach its leader P state, therefore the synchronization of the agent *i* to the leader *P* will be bounded.

Using containment control policy, x_i converges to the convex hull of all the leader's positions.

$$\forall x_i \in \Omega, \Omega = C_i \bigcup C_{-i} : \exists \Upsilon > 0, T\left(\Upsilon, x_i\right) \ni \left\|\delta_i\left(t\right)\right\| < \Upsilon, \forall t \geq T, t_{initial} = 0 \qquad (37)$$

Therefore, the norm of local error dynamic is

$$\left\|\delta_i\right\| = \left\| \sum_{\substack{j=1 \\ j \in C_i}}^{N} a_{ij}(x_j - x_i) + \sum_{\substack{p=1 \\ p \in C_i}}^{P} g_{ip}\left(x_p - x_i\right) + \sum_{\substack{k \in C_{-i} \\ k \neq i}} g_{ik}\left(x_k - x_i\right) + \sum_{k \in C_{-i}} b_{ik}\left(x_k - x_i\right) \right\| < \Upsilon \qquad (38)$$

As a result, we conclude the local synchronization error δ_i of agent *i* is UUB.

Since the system is UUB, there exists a constant α>0 such that

$$\left\|x_j - x_i\right\| < \alpha, \forall i \in N \qquad (39)$$

Remark 7- Theorem 2 proved that the local error δ_i is UUB (Ultimately Uniformly Bounded) $\left\|\delta_i\right\| < \Upsilon$. It is shown in part B, the error dynamic converges to zero which results in δ_i to be asymptotically stable. From the result for stability, it is proved that the global clustering error dynamics is UUB (Ultimately Uniformly Bounded) and ML-MAS is marginally stable which results in the convergence of the agent *i* in cluster *P* to the corresponding leader of the cluster.

Remark 8- The stability of δ_i has been proven in the sequential steps:

Theorem 1 showed ML-MAS is asymptotically stable for all agent *i*.

Using Small Gain theorem in theorem 1 and UUB in theorem 2, if δ_i is asymptotically stable, δ_i will go to zero as time goes to infinity.

However, the convergence of δ_i to zero does not necessarily guarantees the convergence of the agent to its leader.

In the following section, it will be shown how ML-MAS reaches consensus using cluster partitioning techniques.

6 CLUSTER PARTITIONING FOR MULTILEADER MULTI-AGENT SYSTEMS CLUSTER CONSENSUS

In this section, the cluster partitioning techniques will be deployed to reach ML-MAS cluster consensus.

In order for the agents in the same cluster to reach consensus with their leader, a new formulation will be added to the combination of H∞ optimal control policy along with the UUB stability of the dynamic system.

It has been proven that the defined ML-MAS system is UUB (Ultimately Uniformly Bounded) for all agents in the clusters and their corresponding leaders. Since the system has UUB stability, the influence of the leader on the agents from other clusters will be minimal and the link between agents of different clusters can be broken. At this point, the intra-cluster links will be formed and if the link between the neighboring agent and the leader is bigger than a defined limit, the link will be broken to reach Asymptotic Stability.

Once the links are all cut, the agent can only see the other agents in its own cluster.

6.1 MULTILEADER MULTI-AGENT SYSTEM CLUSTER CONSENSUS

In previous section *IV*, the stability of ML-MAS is proved using Small-Gain theorem and Ultimately Uniformly Bounded (UUB). In this part, the cluster consensus of ML-MAS is designed using cooperative tracking technique [41] in order to apply the partitioning.

Assumption 4- The augmented graph *G* contains a spanning tree with at least the root node can get access to the leader node. If graph G is disconnected, each separated subgroup shall be either a single node or contains a spanning tree.

Consider the UUB system designed in theorem 2. Once the agents start following their leader, at some point the influence of the leader to the agents of other cluster weakens, to the point that the leader influence on the agent *j*, $j \in C_{-i}$ becomes minimal and the link between agent *j* and agent *i*, $i \in C_i$ can be broken as they do not follow the same leader for partitioning purposes. At this stage,

the intra-cluster links are formed and the inter-cluster links between the neighboring agent for other clusters and the leader of cluster *P* can be broken to reach asymptotic stability.

Theorem 3- *Cluster Partitioning and Convergence of ML-MAS:* Consider the UUB system designed in theorem 2. There exists constant β such that

$$\exists \beta > 0, \beta > \alpha \ni \left\| x_k - x_i \right\| > \beta : b_{ik} = 0, k \in C_{-i} \quad (40)$$

The inter-cluster link can be broken between agent $k, k \in C_{-i}$ and $i, i \in C_i$.

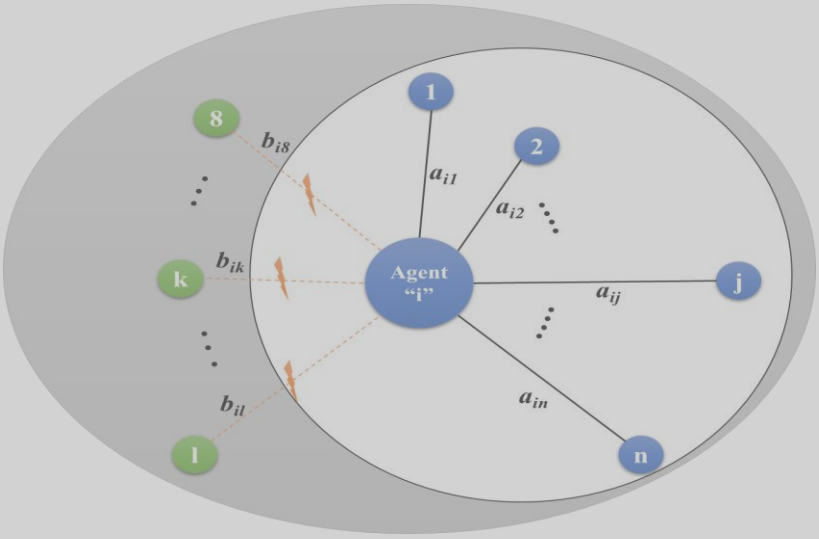


Figure 1- Agent *i* inter-cluster and intra-cluster links

Proof: From theorem 2, assume the system is UUB, there exists a constant α>0 such that

$$\left\| x_j - x_i \right\| < \alpha, \forall i \in N \quad (41)$$

Once the agent *i* from cluster *C* starts following the cluster *C* leader, the influence of the leaders from other clusters -*C* to the agent *i* weakens, to the point that the leader influence on the agent *j*, $j \in C_{-i}$ becomes minimal and the link between agent *j* and agent *i*, $i \in C_i$ can be broken as they do not follow the same leader for partitioning purposes.

At this point, the intra-cluster link will get stronger. If the inter-cluster link between the neighboring agent and cluster *C* leader is bigger than a certain positive indefinite numbers β and ϕ, the link will be broken to reach Asymptotic Stability (A.S.).

For the neighboring agent outside cluster *C*:

$$\exists \beta > 0, \beta > \alpha \ni \left\| x_k - x_i \right\| > \beta : b_{ik} = 0, k \in C_{-i} \tag{42}$$

And for the leaders of other clusters -*C:*

$$\exists \phi > 0, \phi > \alpha \ni \left\| x_p - x_i \right\| > \phi : g_{ip} = 0, p \in C_{-i} \tag{43}$$

Therefore, the synchronization error (3) can be written as

$$\left\| \delta_i \right\| = \left\| \sum_{\substack{j=1 \\ j \in C_i}}^{N} a_{ij} (x_j - x_i) + \sum_{\substack{p=1 \\ p \in C_i}}^{P} g_{ip} \left(x_p - x_i \right) + 0 + 0 \right\| \tag{44}$$

Once all the links are cut, each agent can only see the agent in its own cluster and the cluster partitioning has been accomplished.

Remark 9- *Cluster Consensus of ML-MAS:* The asymptotic stability of the system has been proven using optimal control theory. In theorem 2, UUB theory has been implemented to prove the stability of ML-MAS. And finally, in theorem 3, the cluster partitioning has been deployed to reach cluster consensus for each cluster with its own leader.

6.2 Cluster Consensus of Multileader Multi-Agent Systems using H∞ Optimal Control Policy

Continuous time state feedback control policy guaranties Ultimately Uniformly Boundedness (UUB) for uncertain dynamic systems [37]. Assume the system is described in equation (1) and (2). The uncontrolled system without uncertainty is Lyapunov Stable with respect to the zero state, there exist a class of state feedback controls which are continuous in the state and guarantee that every response of the system is uniformly bounded and uniformly ultimately bounded within the

neighborhood of the zero state. This neighborhood of ultimate boundedness is made arbitrary small [42]. In order to model the minmax differential game and account for variability of the clustering environment, the control law is defined as state feedback control policy system.

Cluster partitioning is deployed to eliminate the effect of the leaders outside the clusters on the agents of the cluster to reach synchronization within the cluster's leader. The analytical results will be represented in section 0.

In the next section, Integral Reinforcement learning will be introduced to learn H∞ optimal control policy solution online to eliminate the dependency of the solution to system dynamics.

7 INTEGRAL REINFORCEMENT LEARNING

In this section, an offline RL algorithm is first given to solve the problem of H∞ optimal design for ML-MAS by learning the solution to the HJI equation optimization.

The performance function is introduced for ML-MAS H∞ optimal problem which is weighted based on the cost of the feedback part of the control input in the performance function. A discount factor upper-bound limit is obtained to assure local asymptotic stability of the error dynamics.

An off-policy IRL algorithm is then developed to learn the solution to the HJI equation online and without requiring any knowledge of the system dynamics [34]. Convergence of this algorithm to the solution to ML-MAS HJI equation is secured.

7.1 INTEGRAL REINFORCEMENT LEARNING FOR OPTIMAL ADAPTIVE CONTROL OF CONTINUOUS-TIME SYSTEMS USING POLICY ITERATION

Reinforcement learning is considerably more complex for continuous-time systems than for discrete-time systems; therefore, fewer results are available on continuous-time system IRL. The development of an offline policy iteration method for continuous-time systems is described in [51]. Using a method known as Integral Reinforcement Learning (IRL) [37], [15] allows the application of reinforcement learning to formulate online optimal adaptive control methods for continuous-time systems. These methods find online solutions to optimal HJI design equations and Riccati equations in real time without knowing the system dynamics *f(x)*, or in the LQR case, without knowing the A matrix.

Consider the continuous-time LQR dynamical system

$$\dot{x} = Ax + Bu \tag{45}$$

$$V\left(x(t)\right) = \frac{1}{2}\int_{i}^{\infty}\left(x^{T}Qx + u^{T}Ru\right)d\tau \tag{46}$$

A policy is called admissible if it is continuous, stabilizes the system, and has a finite associated cost. If the cost is smooth, then an infinitesimal equivalent to the cost function can be found by Leibniz's formula.

In section IV and V, it has been shown that the system is asymptotically stable (A.S.). In this section, H∞ optimal clustering problem will be reformulated to define Off-Policy Integral Reinforcement Learning (IRL) to learn the optimal clustering performance. Consider the system dynamic as defined in section II, equations (1) and (2).

$$\begin{aligned} \dot{x}_i &= Ax_i + Bu_i, i = 1,...,N \\ \dot{x}_p &= Ax_p, p = 1,...,P \end{aligned} \tag{47}$$

The local error dynamics has been defined as δ_i in (3). The fictitious performance output to be controlled is defined such that it satisfies

$$\left\| z_i(t) \right\|^2 = \delta_i^T Q \delta_i + u_i^T R u_i \tag{48}$$

Definition 2 – *Bounded L_2-Gain/Disturbance Attenuation:* The system has a L_2-Gain smaller than ψ where

$$\psi = \min(\gamma_1, \gamma_2) \tag{49}$$

For all local error $\delta_i \in L_2[0,\infty)$:

$$\frac{\int_t^\infty \left\| z_i(\tau) \right\|^2 d\tau}{\int_t^\infty \left\| \delta_i(\tau) \right\|^2 d\tau} \le \psi^2 \tag{50}$$

Where ψ is the amount of attenuation from $\delta_i(t)$ to the defined performance output variable $z_i(t)$. The optimal robust control is minimum of ψ.

The off-policy reinforcement learning algorithm requires some knowledge of the system dynamics. The method presented in this section is to solve H∞ optimal problem with completely unknown dynamics.

7.2 ONLINE ALGORITHM ON AN ACTOR-CRITIC-DISTURBANCE STRUCTURE

In this section we discuss the implementation of the adaptive algorithm on the Actor/Critic/Disturbance structure.

The structure of the system with the adaptive controller is presented as state feedback control model. The policy iteration technique has been adopted to a control system structure that allows the system to perform optimal adoptive control without knowing the internal dynamics of the system. This hybrid solution is constructed as a continuous-time/discrete-time adaptive control structure. This means that the system has continuous-time dynamics, and a discrete-time sampled data portion for policy evaluation.

All algorithm's calculations are performed at a supervisory level which operates based on discrete-time data measured from the system. This optimal clustering intelligent control structure implements the **Policy Iteration** algorithm and uses the Critic neural network to parameterize the performance of the continuous-time control system associated with a certain control policy. The Optimal Clustering Policy structure makes the decisions relative to the discrete-time moments at which the Actor, the Critic, and the disturbance parameters will be updated. The Actor neural network is part of the control system structure and performs continuous-time control, while its constant gain is updated at discrete moments in time. The same algorithm is in play for the disturbance neural network.

The algorithm converges to the solution of the continuous-time optimal control problem, since the Critic update is based on the continuous-time cost over a finite sample interval. The net result is a continuous-time controller incorporated in a continuous-time/discrete-time adaptive structure, which includes the continuous-time dynamics of the cost function and operates based on sampled data, to perform the policy evaluation and policy update steps at discrete moments in time [43].

The cost function solution can be obtained in real time, after enough data points are collected along state trajectories in each cluster. The least-squares method has been adopted for finding the parameters of the cost function. Least-squares method can be replaced with other methods of parameter identification if needed.

The iterations of updating the control policies will be stopped when the error between the system performance evaluated at two consecutive steps will cross below a specified system's threshold. If

this error exceeds the specified threshold, it indicates a change in the system dynamics, which will signal the Critic to start tuning the Actor parameters.

In this algorithm, knowledge of the system dynamics is not required for the cost function and the control policy updates. However, the partial knowledge of the is required for the update of the control policy and this makes the online algorithm partially model free.

In the next section, it is shown how to implement Offline Reinforcement Learning Policy to make the complete model-free algorithm.

7.3 OFF-POLICY IRL FOR LEARNING ML-MAS H∞ OPTIMAL PROBLEM

In this section, an offline RL algorithm is first given to solve ML-MAS H∞ optimal clustering problem by learning the solution to the HJI equation. An off-policy IRL algorithm is then developed to learn the solution to the HJI equation online and without requiring any knowledge of the system dynamics. Actor–Critic–Disturbance structure with three Neural Networks (NNs) are utilized to implement the off-policy IRL algorithm.

7.3.1 OFF-POLICY REINFORCEMENT LEARNING ALGORITHM

The Bellman equation (8) is solved for the cost function *V*. To solve for optimal value function, a Policy Iteration (PI) algorithm iterates on both the control and disturbance players to solve the HJI equation. An offline PI algorithm for solving the H∞ optimal clustering problem is given in Algorithm 1.

Algorithm 1 Offline Reinforcement Learning Algorithm

1. Start
2. Given admissible policy u_0^r
3. For the control policy input u_i, and disturbance policies u_j, and u_k, find cost function V_i using the following Bellman equation:

$$H_i\left(V_i,u_i,u_j,u_k\right)=\delta_i Q_i\delta_i+u_i^T R_i u_i+\nabla V_i^T\left(\delta_i\right)\dot{\delta}_i \\ -\gamma_1^2\sum_{j\in C_i}a_{ij}u_j^T R_j u_j-\gamma_2^2\sum_{k\in C_{-i}}b_{ik}u_k^T R_k u_k=0 \quad (51)$$

4. Update the control policy u_i using

$$u_i^{r+1}=\left(d_i+g_i\right)R_i^{-1}B^T\nabla V \quad (52)$$

5. Update the disturbances policies u_j and u_k using

$$u_j^{r+1}=\frac{1}{\gamma_1^2}R_j^{-1}\nabla V \\ u_k^{r+1}=\frac{1}{\gamma_2^2}R_k^{-1}\nabla V \quad (53)$$

6. Go to 3
7. End.

Algorithm 1 extends the results of the simultaneous RL algorithm in [44] to the optimal clustering problem. The convergence of this algorithm to the minimal non-negative solution of the HJI equation was shown in [44]. The convergence of Algorithm 1 is shown by converging to the unique solution of the HJI equation.

7.3.2 OPTIMAL CLUSTERING CONSENSUS PROBLEM

The next step is to design an optimal control input to make the states of the agents in each cluster $x_i(t)$ follow a desired leader's trajectory of their cluster $x_j(t)$.

Algorithm 1 requires complete knowledge of the system dynamics. Therefore, the off-policy IRL algorithm, which was presented in [45] for solving the *H_2-optimal* regulation problem, is modified to solve the *$H\infty$-optimal* clustering problem for systems with completely unknown dynamics. The system dynamics (4) is written as

$$\dot{\delta}_i = \left[A\delta_i - \left(d_i + g_i\right) Bu_i + \sum_{j\in C_i} a_{ij} Bu_j + \sum_{k\in C_{-i}} b_{ik} Bu_k \right] + \left(d_i + g_i\right) Bu_i^r - \left(d_i + g_i\right) Bu_i^r$$
$$- \sum_{j\in C_i} a_{ij} Bu_j^r + \sum_{j\in C_i} a_{ij} Bu_j^r - \sum_{k\in C_{-i}} b_{ik} Bu_k^r + \sum_{k\in C_{-i}} b_{ik} Bu_k^r \tag{54}$$

where u_i^r, u_j^r, and u_k^r are the policies to be updated. The cost function $V_i(\delta(t))$ is differentiated using algorithm 1 method to solve the following equation

$$\dot{V}\left(\delta(t)\right) = \nabla^T V \dot{\delta}(t) \tag{55}$$

by substituting $\dot{\delta}(t)$ from equation (54), equation (55) is written as

$$\dot{V}\left(\delta(t)\right) = \nabla^T V \left[A\delta_i - \left(d_i + g_i\right) Bu_i^r + \sum_{j\in C_i} a_{ij} Bu_j^r + \sum_{k\in C_{-i}} b_{ik} Bu_k^r \right.$$
$$\left. - \left(d_i + g_i\right) B\left(u_i - u_i^r\right) + \sum_{j\in C_i} a_{ij} B\left(u_j - u_j^r\right) + \sum_{k\in C_{-i}} b_{ik} B\left(u_k - u_k^r\right) \right] \tag{56}$$

In the tracking problem [30], the control input is defined in two terms: a feedforward term that guarantees tracking and a feedback term that stabilizes the system. The feedforward term is obtained using the dynamics inversion concept and the feedback input is found by applying the stationarity condition that is derived from the cost function. Obtaining the feedforward part of the control input needs complete knowledge of the system dynamics and the reference trajectory dynamics. In [46] and [47], a new formulation is developed that gives both feedback and feedforward parts of the control input simultaneously and thus enables RL algorithms to solve the tracking problems without requiring the complete knowledge of the system dynamics.

Remark 10 – *Discounted Performance Function:* Since the reference trajectory does not go to zero in the case of most real applications. The control input contains a feedforward part that depends on the reference trajectory and thus $u_i^T R_i u_i$ does not go to zero as time goes to infinity. Therefore, it is essential to use a discounted performance function for the proposed formulation.

$$\dot{V}\left(\delta(t)\right)=\nabla^T V\left[A\delta_i-\left(d_i+g_i\right)Bu_i^r\right]-\nabla^T V\left(d_i+g_i\right)BR_i^{-1}R_i\left(u_i-u_i^r\right)$$
$$+\nabla^T V\sum_{j\in C_i}a_{ij}Bu_j^r+\frac{\gamma_1^2}{\gamma_1^2}\sum_{j\in C_i}a_{ij}\nabla^T VBR_j^{-1}R_j\left(u_j-u_j^r\right) \tag{57}$$
$$+\nabla^T V\sum_{k\in C_{-i}}b_{ik}Bu_k^r+\frac{\gamma_2^2}{\gamma_2^2}\nabla^T V\sum_{k\in C_{-i}}b_{ik}\nabla^T VR_k^{-1}R_kB\left(u_k-u_k^r\right)$$

The off-policy learning algorithm for the synchronization of ML-MAS that does not require any knowledge of the dynamics is structured as follow:

a. Off-policy Bellman equations are first derived.
b. An actor-critic-disturbance neural network (NN) structure is used to evaluate the value function and find an improved control policy for each agent.
c. An iterative off-policy RL algorithm is given to learn approximate optimal control policies that make the ML-MAS reach consensus and meanwhile guarantee the synchronization of all agents to their leaders.
d. In off-policy RL, a behavior policy is applied to the system to generate the data for learning and a different policy, called the target policy, and is evaluated and updated using measured data.

Using algorithm 1, the cost function (57) is written as

$$\dot{V}\left(\delta(t)\right)=\nabla^T V\left[A\delta_i-\left(d_i+g_i\right)Bu_i^r\right]+\nabla^T V\sum_{j\in C_i}a_{ij}Bu_j^r+\nabla^T V\sum_{k\in C_{-i}}b_{ik}Bu_k^r$$
$$-u_i^{r+1,T}R_i\left(u_i-u_i^r\right)+\gamma_1^2\sum_{j\in C_i}a_{ij}u_j^{r+1,T}R_j\left(u_j-u_j^r\right)+\gamma_2^2\sum_{k\in C_{-i}}b_{ik}u_k^{r+1,T}R_k\left(u_k-u_k^r\right) \tag{58}$$

Since Bellman equation is linear in the cost function V, solving Bellman equation for V is more efficient than solving HJI for V^*.

$$\delta_iQ_i\delta_i+u^r{}_i^TR_iu^r{}_i-\gamma_1^2\sum_{j\in C_i}a_{ij}u^r{}_j^TR_ju^r{}_j-\gamma_2^2\sum_{k\in C_{-i}}b_{ik}u^r{}_k^TR_ku^r{}_k+\nabla V_i^T\left(\delta_i\right)\dot{\delta}_i=0 \tag{59}$$

where $\gamma_1^2>0$ and $\gamma_2^2>0$. The term $\nabla V_i^T\left(\delta_i\right)\dot{\delta}_i$ is extracted from Bellman equation (59) as below

$$\nabla V_i^T\left(\delta_i\right)\dot{\delta}_i=-\delta_iQ_i\delta_i-u^r{}_i^TR_iu^r{}_i+\gamma_1^2\sum_{j\in C_i}a_{ij}u^r{}_j^TR_ju^r{}_j+\gamma_2^2\sum_{k\in C_{-i}}b_{ik}u^r{}_k^TR_ku^r{}_k \tag{60}$$

Substituting (60) in discounted performance function $\dot{V}(\delta(t))$ (57) yields

$$\dot{V}\left(\delta(t)\right)=-\delta_i Q_i \delta_i - {u^r}_i^T R_i {u^r}_i + \gamma_1^2 \sum_{j\in C_i} a_{ij} {u^r}_j^T R_j {u^r}_j + \gamma_2^2 \sum_{k\in C_{-i}} b_{ik} {u^r}_k^T R_k {u^r}_k$$
$$-{u_i}^{r+1,T} R_i \left(u_i - u_i^r\right) + \gamma_1^2 \sum_{j\in C_i} a_{ij} {u_j}^{r+1,T} R_j \left(u_j - u_j^r\right) + \gamma_2^2 \sum_{k\in C_{-i}} b_{ik} {u_k}^{r+1,T} R_k \left(u_k - u_k^r\right) \tag{61}$$

Integrating both sides of (61) results in the following off-policy IRL Bellman equation:

$$V^r\left(\delta_i(t+T)\right) - V^r\left(\delta_i(t)\right) = \int_t^{t+T} \left(-\delta_i Q_i \delta_i - {u^r}_i^T R_i {u^r}_i + \gamma_1^2 \sum_{j\in C_i} a_{ij} {u^r}_j^T R_j {u^r}_j + \gamma_2^2 \sum_{k\in C_{-i}} b_{ik} {u^r}_k^T R_k {u^r}_k \right) d\tau$$
$$+ \int_t^{t+T} \left(-{u_i}^{r+1,T} R_i \left(u_i - u_i^r\right) \right) d\tau + \int_t^{t+T} \left(\gamma_1^2 \sum_{j\in C_i} a_{ij} {u_j}^{r+1,T} R_j \left(u_j - u_j^r\right) + \gamma_2^2 \sum_{k\in C_{-i}} b_{ik} {u_k}^{r+1,T} R_k \left(u_k - u_k^r\right) \right) d\tau \tag{62}$$

The off-policy IRL Bellman equation (62) can be solved for the performance function $\dot{V}\left(\delta_i(t)\right)$ and the updated control and disturbance policies ${u_i}^{r+1}$, ${u_j}^{r+1}$, and ${u_k}^{r+1}$ simultaneously for a fixed control policy u (the policy that is applied to the system) and a given disturbance u_j, and u_k (the actual disturbances are based on the neighbor's policies).

Lemma 2 – *(Off-Policy IRL Bellman Solution)* the solution for performance function solution using the off-policy IRL equation (62) shall be the same the Bellman equation (51), and the updated control and disturbances are the same as (52)-(53).

Proof – The off-policy IRL equation (62) gives the same solution for the performance function as the Bellman equation. By dividing both sides of the off-policy IRL Bellman equation (62) by *T*, and taking limit results in

$$\lim_{T\to 0} \frac{V^r\left(\delta_i(t+T)\right) - V^r\left(\delta_i(t)\right)}{T} + \lim_{T\to 0} \frac{\int_t^{t+T} \left(\delta_i Q_i \delta_i + {u^r}_i^T R_i {u^r}_i - \gamma_1^2 \sum_{j\in C_i} a_{ij} {u^r}_j^T R_j {u^r}_j - \gamma_2^2 \sum_{k\in C_{-i}} b_{ik} {u^r}_k^T R_k {u^r}_k \right) d\tau}{T}$$
$$+ \lim_{T\to 0} \frac{\int_t^{t+T} \left({u_i}^{r+1,T} R_i \left(u_i - u_i^r\right) \right) d\tau}{T} + \lim_{T\to 0} \frac{\int_t^{t+T} \left(-\gamma_1^2 \sum_{j\in C_i} a_{ij} {u_j}^{r+1,T} R_j \left(u_j - u_j^r\right) - \gamma_2^2 \sum_{k\in C_{-i}} b_{ik} {u_k}^{r+1,T} R_k \left(u_k - u_k^r\right) \right) d\tau}{T} = 0 \tag{63}$$

The limits in (63) are the same as the derivative of each limit. Therefore, the terms in (63) are written as follows

$$\lim_{T\to 0} \frac{V^r\left(\delta_i(t+T)\right) - V^r\left(\delta_i(t)\right)}{T} = \dot{V}\left(\delta_i(t)\right) \tag{64}$$

$$\lim_{T\to 0}\frac{\int_{t}^{t+T}\left(\delta_i Q_i \delta_i + u^{r}{}_{i}^{T} R_i u^{r}{}_{i} - \gamma_1^2 \sum_{j\in C_i} a_{ij} u^{r}{}_{j}^{T} R_j u^{r}{}_{j} - \gamma_2^2 \sum_{k\in C_{-i}} b_{ik} u^{r}{}_{k}^{T} R_k u^{r}{}_{k}\right) d\tau}{T} \tag{65}$$

$$= \delta_i Q_i \delta_i + u^{r}{}_{i}^{T} R_i u^{r}{}_{i} - \gamma_1^2 \sum_{j\in C_i} a_{ij} u^{r}{}_{j}^{T} R_j u^{r}{}_{j} - \gamma_2^2 \sum_{k\in C_{-i}} b_{ik} u^{r}{}_{k}^{T} R_k u^{r}{}_{k}$$

$$\lim_{T\to 0}\frac{\int_{t}^{t+T}\left(u_i^{r+1,T} R_i \left(u_i - u_i^{r}\right)\right) d\tau}{T} = u_i^{r+1,T} R_i \left(u_i - u_i^{r}\right) \tag{66}$$

$$\lim_{T\to 0}\frac{\int_{t}^{t+T}\left(-\gamma_1^2 \sum_{j\in C_i} a_{ij} u_j^{r+1,T} R_j \left(u_j - u_j^{r}\right) - \gamma_2^2 \sum_{k\in C_{-i}} b_{ik} u_k^{r+1,T} R_k \left(u_k - u_k^{r}\right)\right) d\tau}{T} \tag{67}$$

$$= -\gamma_1^2 \sum_{j\in C_i} a_{ij} u_j^{r+1,T} R_j \left(u_j - u_j^{r}\right) - \gamma_2^2 \sum_{k\in C_{-i}} b_{ik} u_k^{r+1,T} R_k \left(u_k - u_k^{r}\right)$$

Substituting (64)-(67) in (63) yields

$$\begin{aligned} &\dot{V}\left(\delta_i(t)\right) + \delta_i Q_i \delta_i + u^{r}{}_{i}^{T} R_i u^{r}{}_{i} - \gamma_1^2 \sum_{j\in C_i} a_{ij} u^{r}{}_{j}^{T} R_j u^{r}{}_{j} - \gamma_2^2 \sum_{k\in C_{-i}} b_{ik} u^{r}{}_{k}^{T} R_k u^{r}{}_{k} \\ &+ u_i^{r+1,T} R_i \left(u_i - u_i^{r}\right) - \gamma_1^2 \sum_{j\in C_i} a_{ij} u_j^{r+1,T} R_j \left(u_j - u_j^{r}\right) - \gamma_2^2 \sum_{k\in C_{-i}} b_{ik} u_k^{r+1,T} R_k \left(u_k - u_k^{r}\right) = 0 \end{aligned} \tag{68}$$

Substituting the updated policies u_i^{r+1}, u_j^{r+1}, and u_k^{r+1} from (52)-(53) into (68) gives the Bellman equation (51). This completes the proof.

Remark 11 - In the off-policy IRL Bellman equation (62), the control input *u*, which is applied to the system, can be different from the control policy u_i, which is evaluated and updated. The fixed control policy *u* should be a stable control policy. However, disturbance u_j, and u_k are the disturbances that are evaluated and updated.

Remark 12 - Algorithm 2 has two separate phases based on the off-policy algorithm described in [30].

Phase 1 – A fixed initial control policy *u* is applied, and the system data is recorded. The data is gathered over the time interval *T*.

Phase 2 – In this phase, the data collected in phase 1 is repeatedly used to find a sequence of updated policy u_i and disturbances u_j, and u_k converging to u_i^*, u_j^*, and u_k^*, without requiring any knowledge of the system dynamics. Equation xx is solved using the least-square of collected data samples from the system. It has been shown how to solve the equation (69) for $\dot{V}\left(\delta_i(t)\right)$, and the updated control and disturbance policies u_i^{r+1}, u_j^{r+1}, and u_k^{r+1} simultaneously. After the learning is done and the optimal control policy u^* is found, it can then be applied to the system.

Algorithm 2 Online Off-Policy RL Algorithm for Solving Optimal Clustering HJI Equation

1. Start
2. Given admissible policy u_0^r
3. For *N* different sampling interval *T*
4. For the control policy input u_i, and disturbance policies u_j, and u_k collect required system data:
5. ***Phase 1 (Data collection of the targeted variables using a fixed control policy):*** Collect System state, control input and disturbance
6. ***Phase 2 (Data regression of collected data sequentially to find an optimal policy iteratively):*** Use collected data in phase 1 to solve the following Bellman equation for $\dot{V}\left(\delta_i(t)\right)$, and the updated control and disturbance policies u_i^{r+1}, u_j^{r+1}, and u_k^{r+1} simultaneously:

$$
\begin{aligned}
&V^r\left(\delta_i(t+T)\right)-V^r\left(\delta_i(t)\right)= \\
&\int_t^{t+T}\left(-\delta_i Q_i \delta_i - {u^r}_i^T R_i {u^r}_i + \gamma_1^2 \sum_{j\in C_i} a_{ij} {u^r}_j^T R_j {u^r}_j + \gamma_2^2 \sum_{k\in C_{-i}} b_{ik} {u^r}_k^T R_k {u^r}_k\right) d\tau \\
&+\int_t^{t+T}\left(-{u_i}^{r+1,T} R_i\left(u_i-u_i^r\right)\right) d\tau \\
&+\int_t^{t+T}\left(\gamma_1^2 \sum_{j\in C_i} a_{ij} {u_j}^{r+1,T} R_j\left(u_j-u_j^r\right)+\gamma_2^2 \sum_{k\in C_{-i}} b_{ik} {u_k}^{r+1,T} R_k\left(u_k-u_k^r\right)\right) d\tau
\end{aligned}
\tag{69}
$$

7. Stop if a stopping criterion is met, otherwise set *i = i +1* and go to 6.
8. End.

Theorem 4- (Convergence of ML-MAS and Cluster Consensus): The off-policy Algorithm 2 converges to the optimal control and disturbance solutions given by (52)-(53),

$$u_i^{r+1} = \left(d_i + g_i\right) R_i^{-1} B^T \nabla V$$

$$u_j^{r+1} = \frac{1}{\gamma_1^2} R_j^{-1} \nabla V$$

$$u_k^{r+1} = \frac{1}{\gamma_2^2} R_k^{-1} \nabla V$$

where the value function satisfies the Optimal Clustering HJI equation.

Proof: It was shown in Lemma 2 that the off-policy optimal clustering Bellman equation (69) gives the same value function as the Bellman equation (51) and the same updated policies as (52) -(53). Therefore, both Algorithms 1 and 2 have the same convergence properties. Convergence of Algorithm 1 is proved in [44]. This confirms that Algorithm 2 converges to the optimal solution.

Remark 12 – It is important to mention that although both Algorithms 1 and 2 have the same convergence properties, Algorithm 2 finds an optimal control policy without requiring any knowledge of the system dynamics. This contrasts with algorithm 1 that requires full knowledge of the system dynamics. Moreover, Algorithm 1 is an on-policy RL algorithm, which requires the disturbance input to be specified and adjustable. On the other hand, Algorithm 2 is an off-policy RL algorithm, which eliminates this requirement.

7.4 ML-MAS Off-Policy IRL using Neural Networks

In this section, the ML-MAS off-policy RL Algorithm 2 is implemented by utilizing the collected data found by applying a fixed control policy *u* to the system to solve (69) for $\dot{V}_i\left(\delta_i(t)\right)$, and the updated control and disturbance policies u_i^{r+1}, u_j^{r+1}, and u_k^{r+1} iteratively. Three NNs, i.e., the actor NN, the critic NN, and the disturber NN, are used here to approximate the value function and the updated control and disturbance policies in the Bellman equation (69). The solution $\dot{V}\left(\delta_i(t)\right)$, and the updated control and disturbance policies u_i^{r+1}, u_j^{r+1}, and u_k^{r+1} of the Bellman equation (69) are approximated by three NNs as

$$\hat{V}_i\left(\delta_i(t)\right)=\hat{W}_1^T\sigma\left(\delta_i(t)\right) \tag{70}$$

$$\hat{u}_i^{r+1}\left(\delta_i(t)\right)=\hat{W}_2^T\varepsilon\left(\delta_i(t)\right) \tag{71}$$

$$\hat{u}_j^{r+1}\left(\delta_i(t)\right)=\hat{W}_3^T\varphi\left(\delta_i(t)\right) \tag{72}$$

$$\hat{u}_k^{r+1}\left(\delta_i(t)\right)=\hat{W}_4^T\kappa\left(\delta_i(t)\right) \tag{73}$$

where $\sigma=\left[\sigma_1,...,\sigma_{l_1}\right]\in\mathbb{R}^{l_1}$, $\varepsilon=\left[\varepsilon_1,...,\varepsilon_{l_2}\right]\in\mathbb{R}^{l_2}$, $\varphi=\left[\varphi_1,...,\varphi_{l_3}\right]\in\mathbb{R}^{l_3}$, and $\kappa=\left[\kappa_1,...,\kappa_{l_4}\right]\in\mathbb{R}^{l_4}$ provide appropriate basis function vectors, $\hat{W}_1^T\in\mathbb{R}^{l_1}$, $\hat{W}_2^T\in\mathbb{R}^{m\times l_2}$, $\hat{W}_3^T\in\mathbb{R}^{q\times l_3}$, and $\hat{W}_4^T\in\mathbb{R}^{s\times l_4}$. l_1, l_2, l_3, and l_4 are the number of neurons.

Define $\upsilon^1=\left[\upsilon_1^1,...,\upsilon_m^1\right]=u-u_i$, $\upsilon^2=\left[\upsilon_1^2,...,\upsilon_q^2\right]=u-u_j$, $\upsilon^3=\left[\upsilon_1^3,...,\upsilon_s^3\right]=u-u_k$ and assume $R_i=Diag\left(r_{i,1},...,r_{i,m}\right)$, $R_j=Diag\left(r_{j,1},...,r_{j,q}\right)$, and $R_k=Diag\left(r_{k,1},...,r_{k,s}\right)$.

Using the above assumptions and definitions and substituting (70)-(73) in the Bellman equation (69), the **Bellman Approximation Error** $e(t)$ is derived as

$$\begin{aligned}
&e(t)=\hat{W}_1^T\left[\sigma\left(\delta_i(t+T)\right)-\sigma\left(\delta_i(t)\right)\right]\\
&+\int_t^{t+T}\left(\delta_iQ_i\delta_i+u^{r}{}_i^T R_i u^{r}{}_i-\gamma_1^2\sum_{j\in C_i}a_{ij}u^{r}{}_j^T R_j u^{r}{}_j-\gamma_2^2\sum_{k\in C_{-i}}b_{ik}u^{r}{}_k^T R_k u^{r}{}_k\right)d\tau\\
&+\int_t^{t+T}\left(\hat{W}_{2,l}^T\varepsilon\left(\delta_i(t)\right)R_i\upsilon_m^1\right)d\tau-\int_t^{t+T}\left(\gamma_1^2\sum_{j\in C_i}a_{ij}\hat{W}_{3,k}^T\varphi\left(\delta_i(t)\right)R_j\upsilon_q^2+\gamma_2^2\sum_{k\in C_{-i}}b_{ik}\hat{W}_{4,p}^T\kappa\left(\delta_i(t)\right)R_k\upsilon_s^3\right)d\tau
\end{aligned} \tag{74}$$

where $\hat{W}_{2,l}$ is the l^{th} column of $\hat{W}_2$, $\hat{W}_{3,k}$ is the k^{th} column of $\hat{W}_3$, and $\hat{W}_{4,p}$ is the l^{th} column of $\hat{W}_4$. The Bellman approximation error is the continuous-time counterpart of the Temporal Difference (TD) [48]. The least-squares method is used to minimize the value of the temporal difference. The Bellman Approximation Error (74) is rewritten as

$$y(t)+e(t)=\hat{W}^T h(t) \tag{75}$$

where $\hat{W}$, $h(t)$, and $y(t)$ are

$$\hat{W} = \left[\hat{W}_1^T, \hat{W}_{2,l}^T, ..., \hat{W}_{2,m}^T, \hat{W}_{3,l}^T, ..., \hat{W}_{3,q}^T, \hat{W}_{4,l}^T, ..., \hat{W}_{4,s}^T \right]$$
$$\hat{W} \in \mathbb{R}^{l_1 + m \times l_2 + q \times l_3 + s \times l_4} \tag{76}$$

$$h(t) = \begin{bmatrix} \sigma\left(\delta_i(t+T)\right) - \sigma\left(\delta_i(t)\right) \\ r_{i,1} \int_t^{t+T} \left(\hat{W}_{2,l}^T \varepsilon\left(\delta_i(t)\right) \upsilon_1^1 \right) d\tau \\ \vdots \\ r_{i,m} \int_t^{t+T} \left(\hat{W}_{2,l}^T \varepsilon\left(\delta_i(t)\right) \upsilon_m^1 \right) d\tau \\ -\int_t^{t+T} \left(\gamma_1^2 \sum_{j \in C_i} a_{ij} \hat{W}_{3,k}^T \varphi\left(\delta_i(t)\right) R_j \upsilon_1^2 \right) d\tau \\ \vdots \\ -\int_t^{t+T} \left(\gamma_1^2 \sum_{j \in C_i} a_{ij} \hat{W}_{3,k}^T \varphi\left(\delta_i(t)\right) R_j \upsilon_q^2 \right) d\tau \\ -\int_t^{t+T} \left(\gamma_2^2 \sum_{k \in C_{-i}} b_{ik} \hat{W}_{4,p}^T \kappa\left(\delta_i(t)\right) R_k \upsilon_1^3 \right) d\tau \\ \vdots \\ -\int_t^{t+T} \left(\gamma_2^2 \sum_{k \in C_{-i}} b_{ik} \hat{W}_{4,p}^T \kappa\left(\delta_i(t)\right) R_k \upsilon_s^3 \right) d\tau \end{bmatrix} \tag{77}$$

$$y(t) = \int_t^{t+T} \left(\delta_i Q_i \delta_i + u^{r}{}_i^T R_i u^{r}{}_i \right) d\tau - \int_t^{t+T} \left(\gamma_1^2 \sum_{j \in C_i} a_{ij} u^{r}{}_j^T R_j u^{r}{}_j + \gamma_2^2 \sum_{k \in C_{-i}} b_{ik} u^{r}{}_k^T R_k u^{r}{}_k \right) d\tau \tag{78}$$

The parameter vector $\hat{W}$, which gives the approximated value function, actor, critic, and disturbances (70)-(73) is found by minimizing the Bellman approximation error (75) using the least-square method. Assume that the systems state, input, and disturbances data are collected at $N \geq l_1 + m \times l_2 + q \times l_3 + s \times l_4$ (the number of independent elements in vector $\hat{W}$) points t_1 to t_N in the state space, over the same time interval T in *phase 1*. Then, for a given control policy u_i and disturbance u_j, and u_k, (76) and (77) is evaluated at N points to form

$$H = \left[h(t_1), \ldots, h(t_N) \right] \tag{79}$$

$$Y = \left[y(t_1), \ldots, y(t_N) \right]^T \tag{80}$$

The least-square solution to (75) is

$$\hat{W} = \left(HH^T \right)^{-1} HY \tag{81}$$

Which results in the solutions for $\dot{V}\left(\delta_i(t)\right)$, and the updated control and disturbance policies u_i^{r+1}, u_j^{r+1}, and u_k^{r+1}.

Remark 13: Although $\delta_i(t+T)$ appears in the Bellman approximation error (74), this equation is solved using least-square method after observing N samples $\delta_i(t), \delta_i(t+T), \ldots, \delta_i(t+NT)$. Therefore, the knowledge of the system dynamic is not required to predict the future state $\delta_i(t+T)$ at time *t* to solve (74).

8 SIMULATION

In this section, first, the proposed cluster partitioning techniques will be deployed to reach ML-MAS cluster consensus. Then, the proposed off-policy IRL method is applied to the same system to show that it converges to the optimal solution without the knowledge of system dynamics.

8.1 CLUSTER PARTITIONING OF MULTILEADER MULTI-AGENT SYSTEMS WITH (9) AGENTS AND (3) LEADERS

Consider an ML-MAS of $N = 9$ agents and $P = 3$ leaders with the system dynamics as described in equation (1) and (2). The graph topology is shown in Figure **2**. The agents are partitioned into three clusters:

$C_1 = \{1, 2, 3\},$
$C_2 = \{4, 5, 6\},$
$C_3 = \{7, 8, 9\}.$

System matrices of these agents are

$$A = A_i = \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix}, i = 1, \ldots, 9$$
$$B = B_i = \begin{bmatrix} 2 & +1 \\ 1 & 2 \end{bmatrix}, i = 1, \ldots, 9 \quad (82)$$

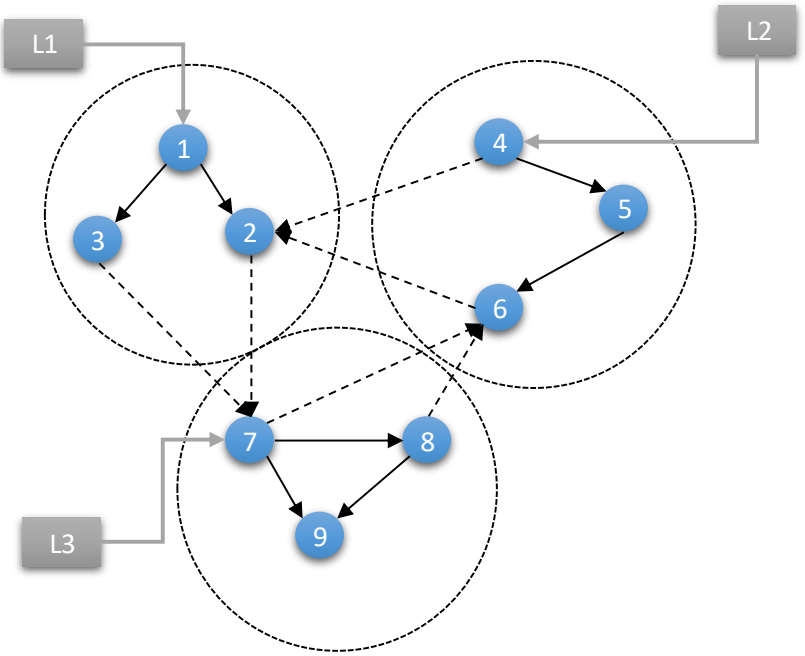


Figure 2- ML-MAS Graph Topology

And the graph topology matrix G is defined as

$$G=\left[G_{ij}\right], i=1,...,9,\ j=1,...,9 \tag{83}$$

where G_{ij} are defined as follow

$$\begin{aligned} G_{11} &= \begin{bmatrix} 0 & 0 & 0 \\ 1 & 0 & 0 \\ 1 & 0 & 3 \end{bmatrix}, G_{22} = \begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}, G_{33} = \begin{bmatrix} 0 & 0 & 3 \\ 1 & 0 & 0 \\ 1 & 1 & 0 \end{bmatrix} \\ G_{12} &= \begin{bmatrix} 0 & 0 & 0 \\ 1 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix}, G_{23} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 1 & 1 & 0 \end{bmatrix}, G_{31} = \begin{bmatrix} 0 & 1 & 1 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \\ G_{13} &= G_{21} = G_{32} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \end{aligned} \tag{84}$$

The results for the theorem 1 are shown in Figure 3 - ML-MAS Synchronization – No UUB - Output trajectories of agents in each cluster - All Agents Synchronization - No cluster partitioning

Figure 4.

Theorem 1 results show ML-MAS is asymptotically stable for all agent *i*. The agents in each cluster synchronize with their leader. However, the convergence of δ_i to zero does not necessarily guarantees the convergence of the agent to its leader and cluster consensus.

By applying theorem 2 and theorem 3, It is shown ML-MAS reaches consensus using cluster partitioning techniques. Once the agent *i* from cluster *C* starts following the cluster *C* leader, the influence of the leaders from other clusters -*C* to the agent *i* weakens, to the point that the leader influence on the agent *j*, $j \in C_{-i}$ becomes minimal and the link between agent *j* and agent *i*, $i \in C_i$ can be broken as they do not follow the same leader for partitioning purposes.

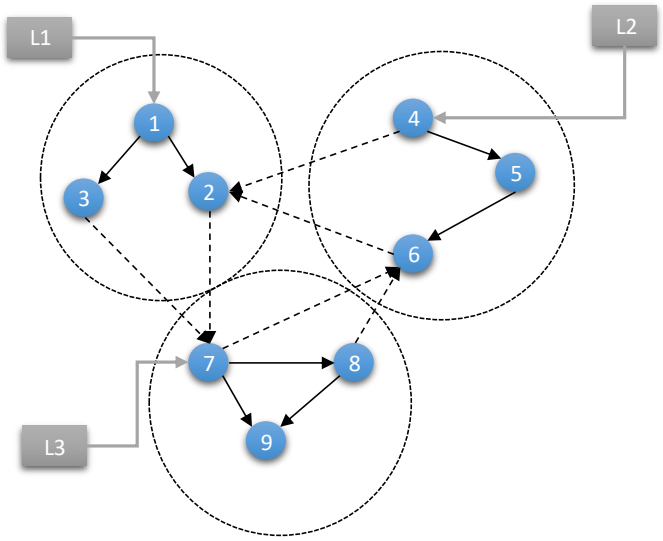
The graph topology matrix *G* is modified once the links are broken as follow

$$G_{11} = \begin{bmatrix} 0 & 0 & 0 \\ 1 & 0 & 0 \\ 1 & 0 & 3 \end{bmatrix}, G_{22} = \begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}, G_{33} = \begin{bmatrix} 0 & 0 & 3 \\ 1 & 0 & 0 \\ 1 & 1 & 0 \end{bmatrix}$$
$$G_{ij} = \gamma \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \ \forall i, j \ \ni i \neq j \tag{85}$$

The results are shown in The ML-MAS off-policy IRL Algorithm 2 is implemented by utilizing the collected data found by applying a fixed control policy *u* to the system to solve the Bellman equation for $V_i(\delta_i(t))$, and the updated control and disturbance policies u_i^{r+1} , u_j^{r+1} , and u_k^{r+1} iteratively. Three NNs, i.e., the actor NN, the critic NN, and the disturber NN, is used to approximate the cost function and the updated control and disturbance policies in the Bellman

equation. The solution $V_i(\delta_i(t))$, and the updated control and disturbance policies is approximated by three NNs.

Figure 11. The asymptotic stability of the system has been reached using optimal control theory. By implementing theorem 2, UUB theory has been implemented and the stability of ML-MAS has been reached. And finally, using theorem 3, the cluster partitioning has been deployed and cluster consensus is reached for each cluster with its own leader.



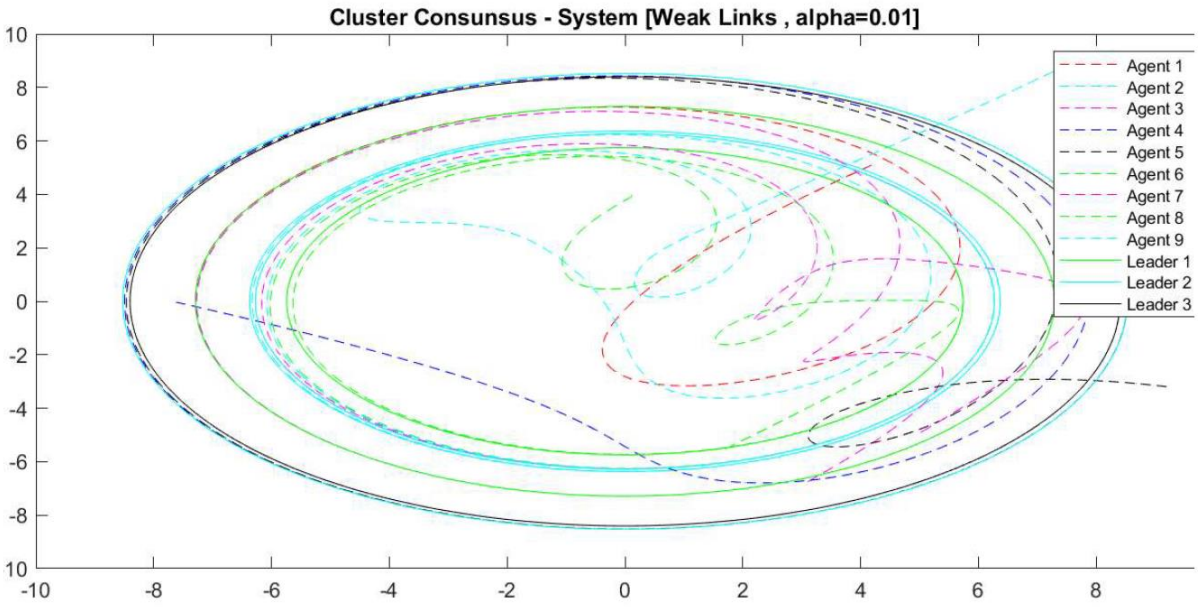


Figure 3 - ML-MAS Synchronization – No UUB - Output trajectories of agents in each cluster - All Agents Synchronization - No cluster partitioning

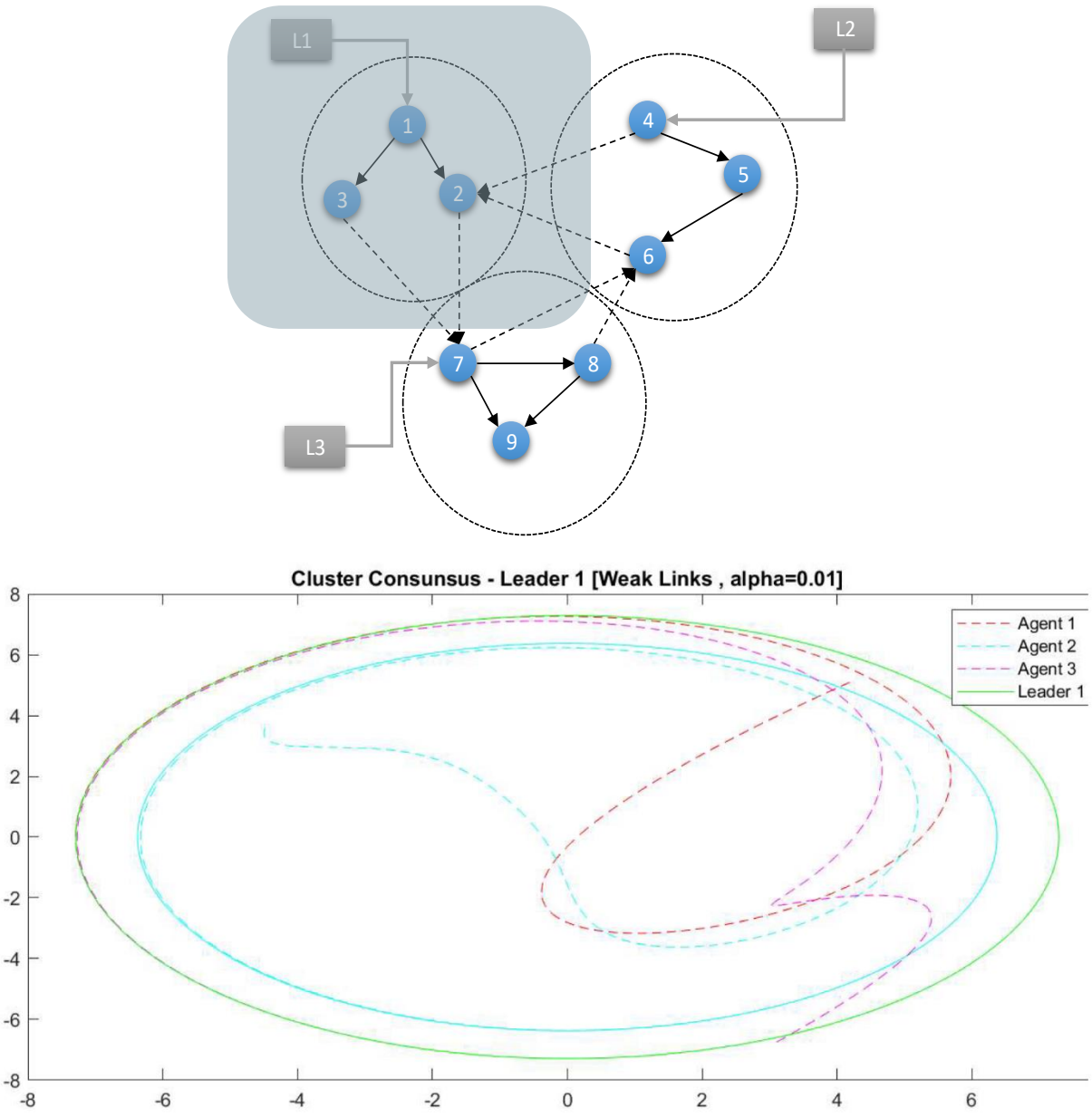


Figure 4 – ML-MAS Synchronization – No UUB - Output trajectories of agents in each cluster - Cluster 1 Synchronization - No cluster partitioning

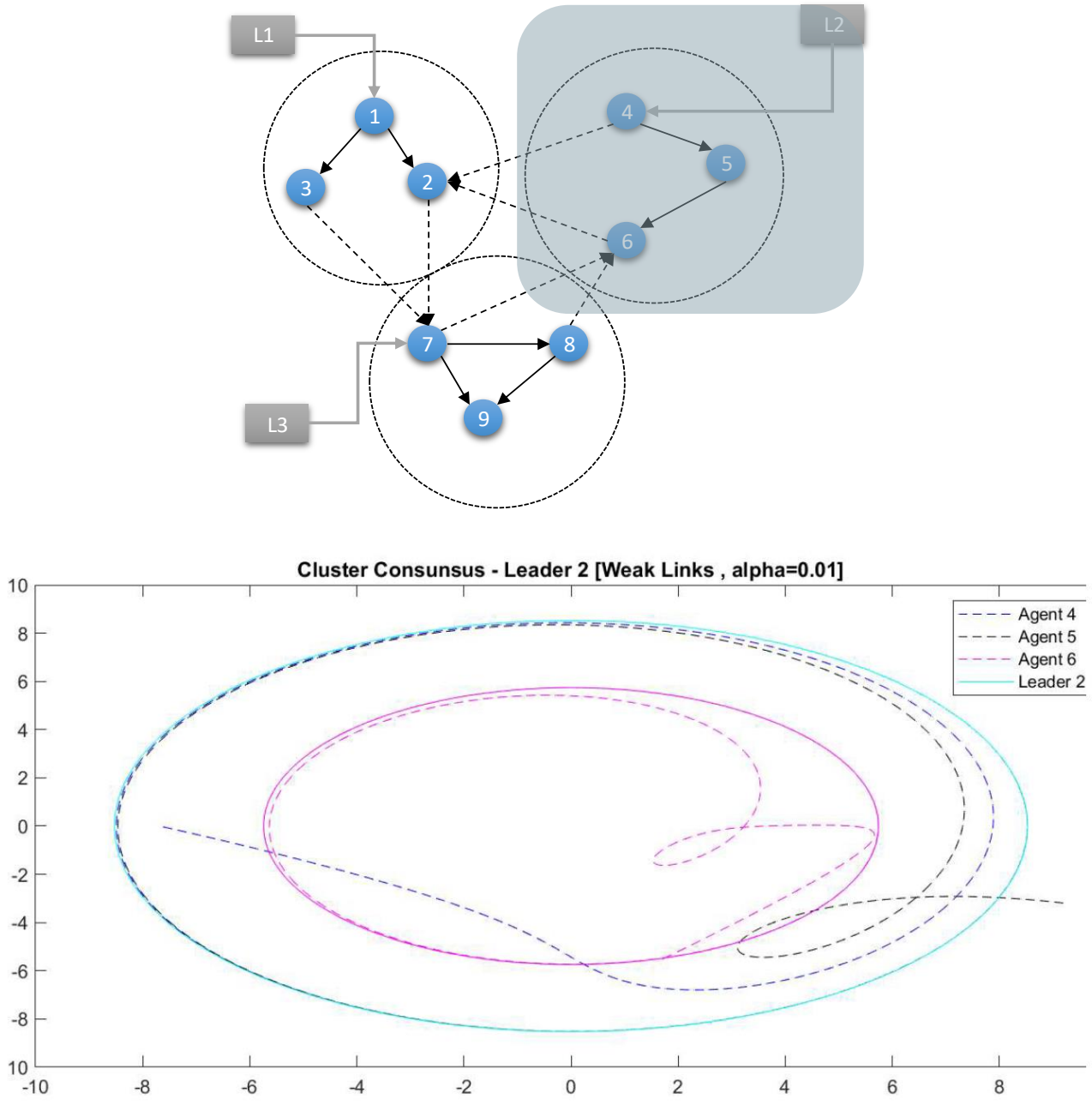


Figure 5 – ML-MAS Synchronization – No UUB - Output trajectories of agents in each cluster - Cluster 2 Synchronization - No cluster partitioning

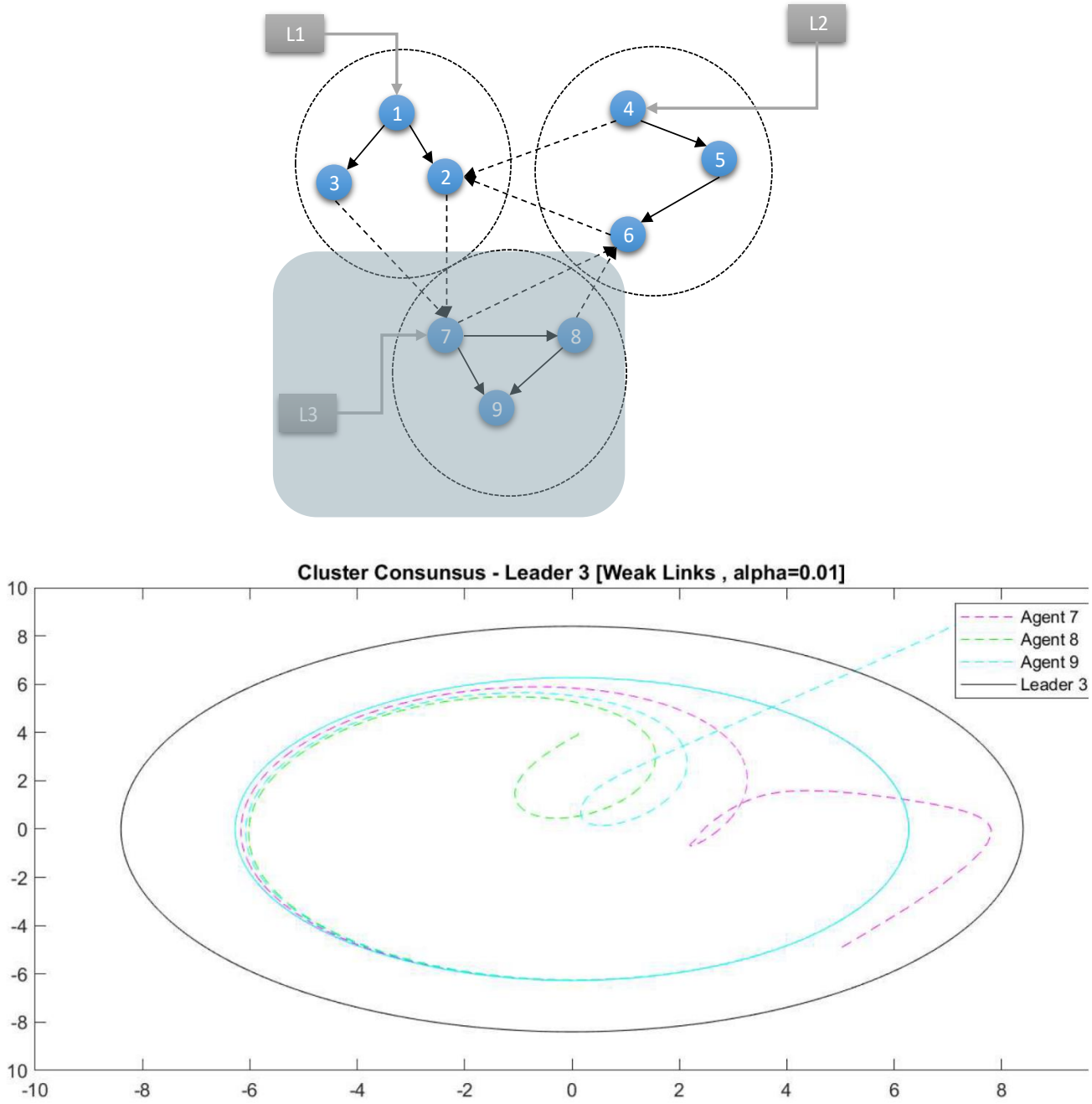


Figure 6 – ML-MAS Synchronization – No UUB - Output trajectories of agents in each cluster - Cluster 3 Synchronization - No cluster partitioning

8.2 Cluster Partitioning of Multileader Multi-agent Systems with UUB Stability and Cluster Partitioning

By applying *theorem 2 and theorem 3*,

Theorem 2- UUB Stability of ML-MAS

Theorem 3- Cluster Partitioning and Convergence of ML-MAS,

It is shown ML-MAS reaches consensus using cluster partitioning techniques. Once the agent *i* from cluster *C* starts following the cluster *C* leader, the influence of the leaders from other clusters -*C* to the agent i weakens, to the point that the leader influence on the agent *j, j* $\subset C_{-i}$ becomes minimal and the link between agent *j* and agent *i, i* $\subset C_i$ can be broken as they do not follow the same leader for partitioning purposes. The graph topology matrix *G* is modified once the links are broken.

$$G_{11} = \begin{bmatrix} 0 & 0 & 0 \\ 1 & 0 & 0 \\ 1 & 0 & 3 \end{bmatrix}, G_{22} = \begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}, G_{33} = \begin{bmatrix} 0 & 0 & 3 \\ 1 & 0 & 0 \\ 1 & 1 & 0 \end{bmatrix}$$
$$G_{ij} = \gamma \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \ \forall i, j \ \ni i \neq j \tag{86}$$

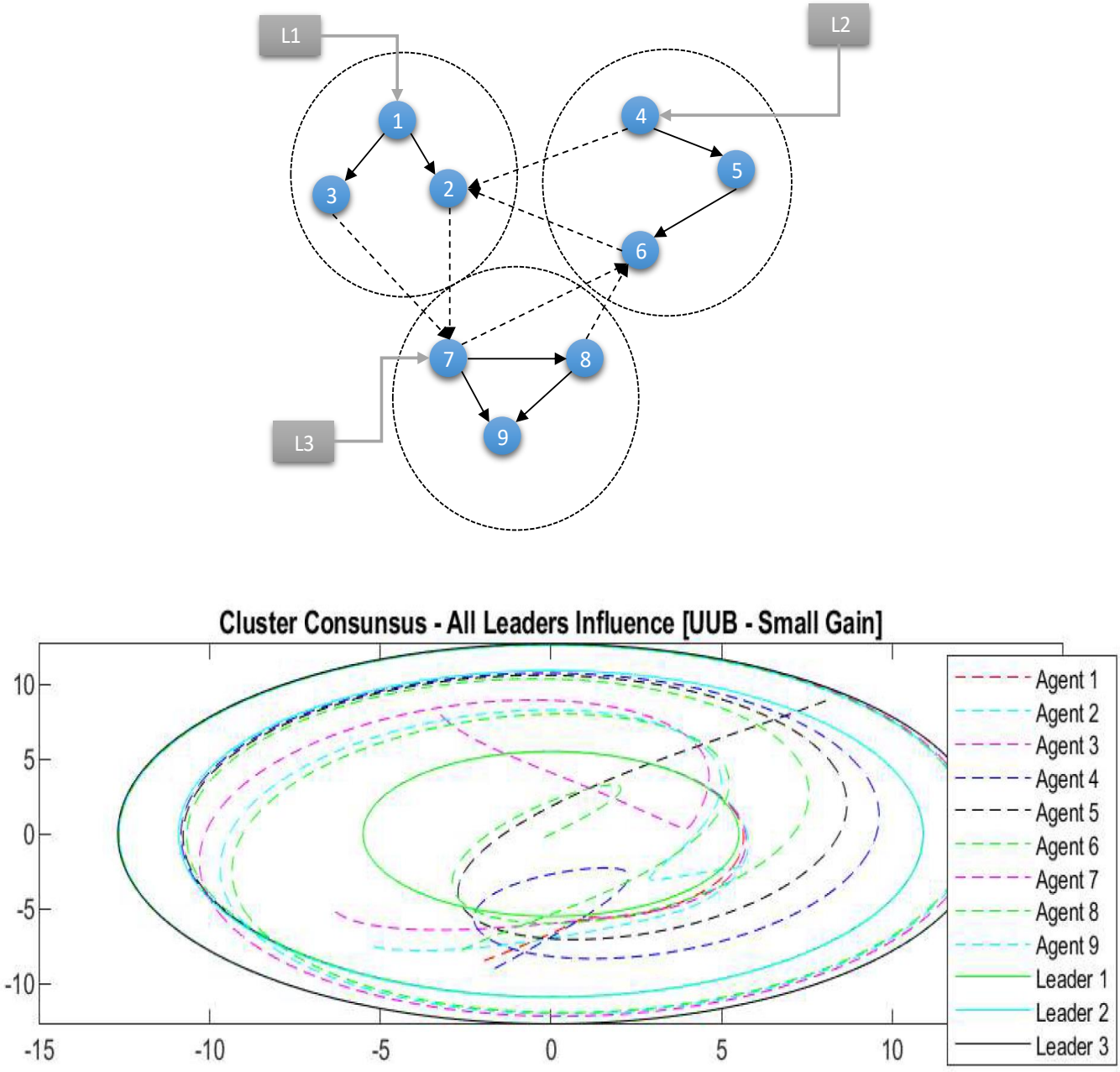


Figure 7 – ML-MAS Synchronization –UUB + Small Gain - Output trajectories of agents in each cluster – All Agents Consensus

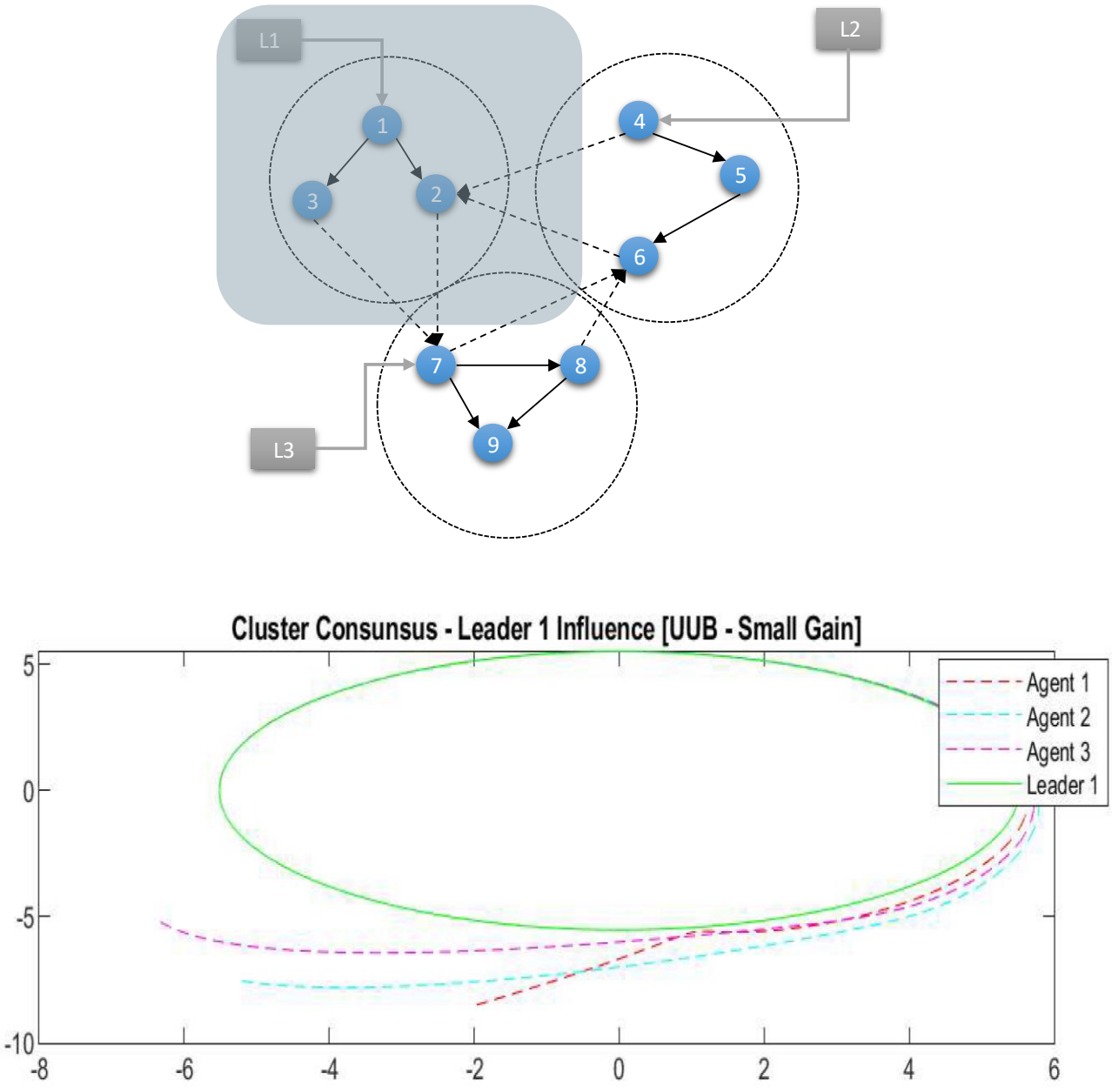


Figure 8 – ML-MAS Synchronization –UUB + Small Gain - Output trajectories of agents in each cluster – Cluster 1 Consensus

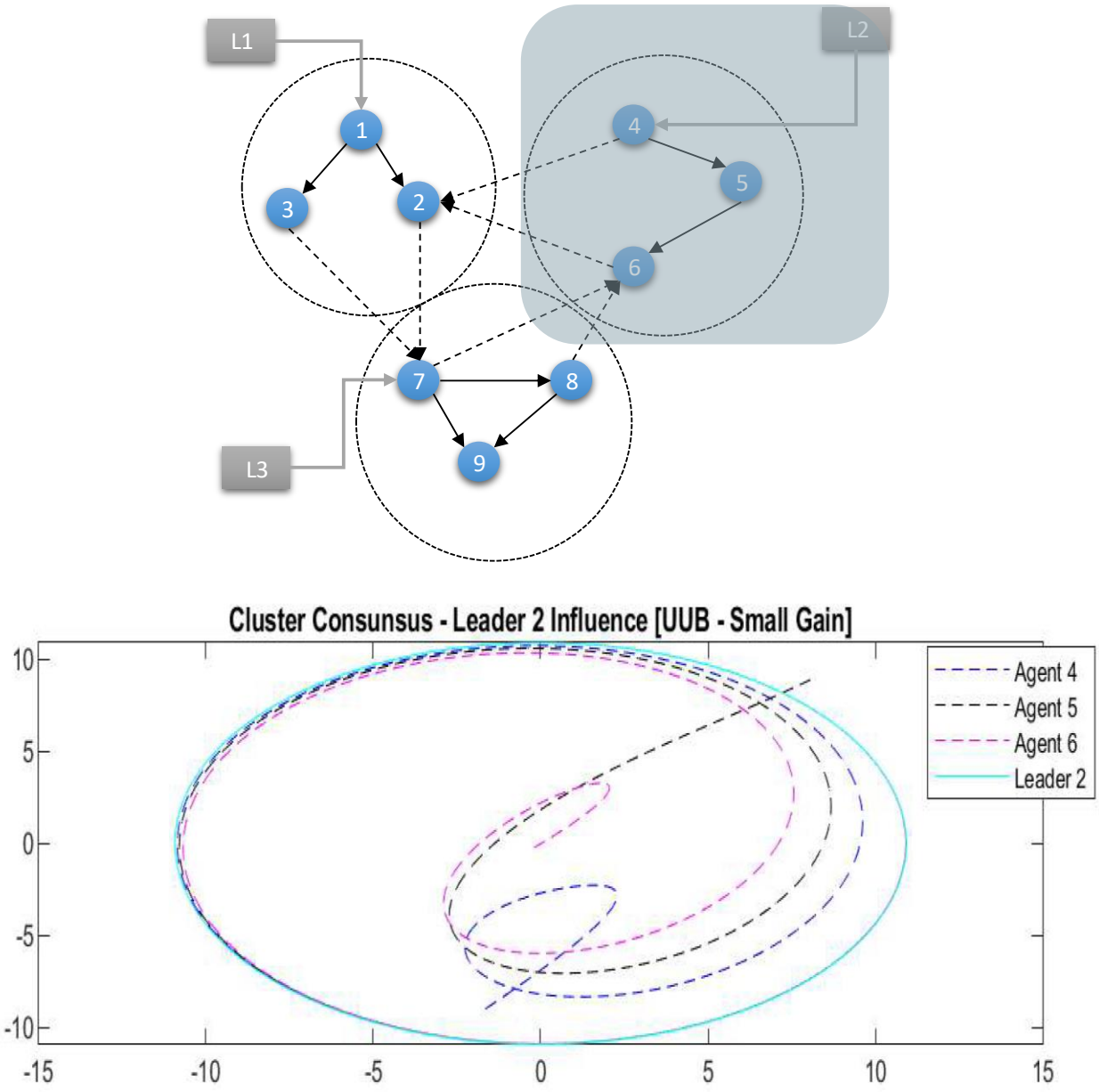


Figure 9 – ML-MAS Synchronization –UUB + Small Gain - Output trajectories of agents in each cluster – Cluster 2 Consensus

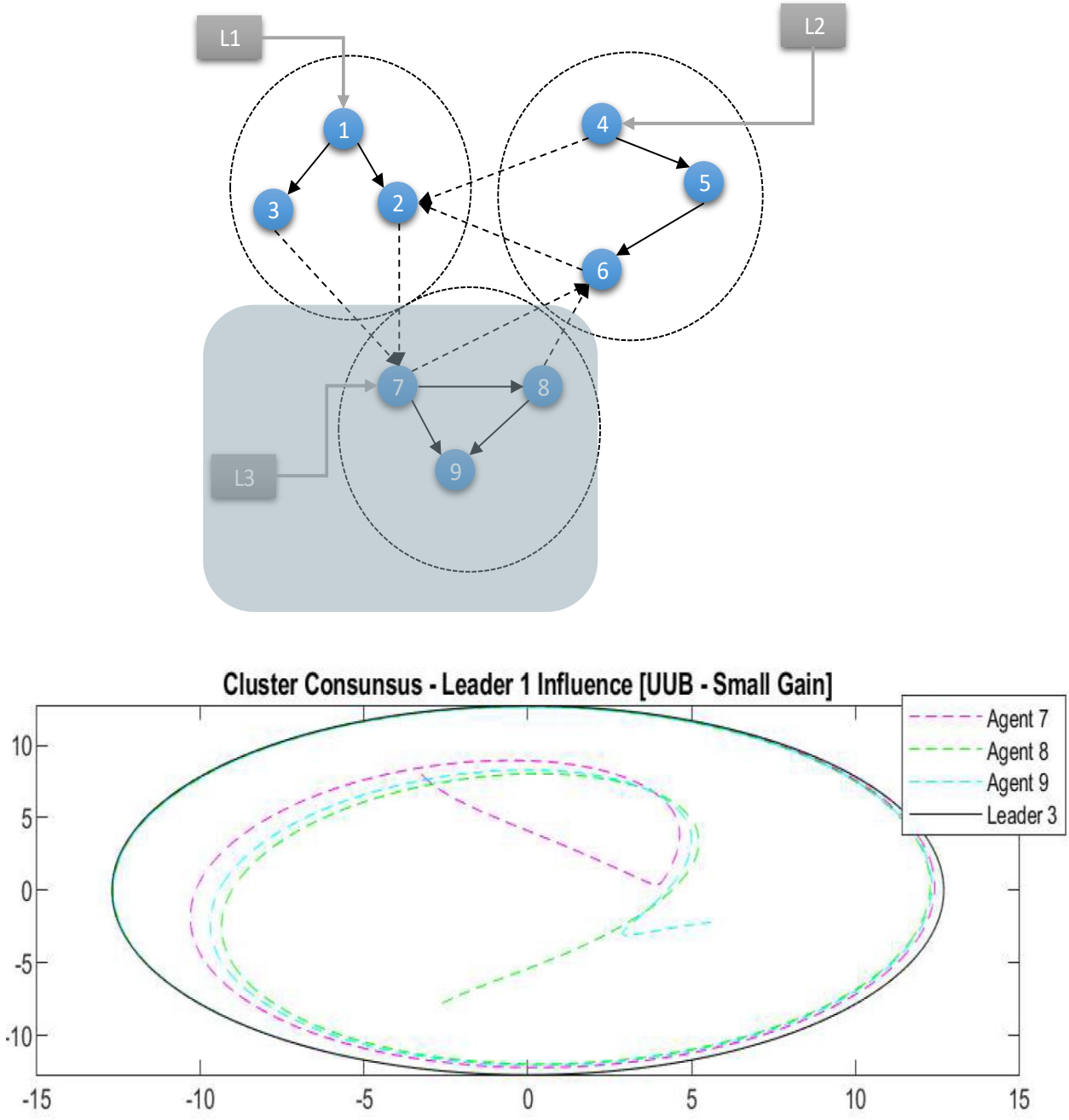


Figure 10 – ML-MAS Synchronization –UUB + Small Gain - Output trajectories of agents in each cluster – Cluster 3 Consensus

The asymptotic stability of the system has been reached using optimal control theory. By implementing theorem 2, UUB theory has been implemented and the stability of ML-MAS has been reached.

And finally, using theorem 3, the cluster partitioning has been deployed and cluster consensus is reached for each cluster with its own leader.

While ML-MAS system is stable, there always exists a solution $P_i>0$ such that the GARE equation holds. The results for a random initial are as follows.

While ML-MAS system is stable, there always exists a solution $P_i > 0$ such that the GARE equation (14) holds. The $P_i > 0$ results for a random initial are as follows.

$$
\begin{aligned}
&P_1 = \begin{bmatrix} 0.4828 & -0.2264 \\ -0.2264 & 0.7028 \end{bmatrix}, P_2 = \begin{bmatrix} 0.4823 & -0.2263 \\ -0.2263 & 0.7020 \end{bmatrix}, P_3 = \begin{bmatrix} 0.4756 & -0.2246 \\ -0.2246 & 0.6896 \end{bmatrix} \\
&P_4 = \begin{bmatrix} 0.8170 & -0.2457 \\ -0.2457 & 1.2785 \end{bmatrix}, P_5 = \begin{bmatrix} 0.4756 & -0.2246 \\ -0.2246 & 0.6896 \end{bmatrix}, P_6 = \begin{bmatrix} 0.4823 & -0.2263 \\ -0.2263 & 0.7020 \end{bmatrix} \\
&P_7 = \begin{bmatrix} 1.5667 & -0.1827 \\ -0.1827 & 2.2330 \end{bmatrix}, P_8 = \begin{bmatrix} 0.4756 & -0.2246 \\ -0.2246 & 0.6896 \end{bmatrix}, P_9 = \begin{bmatrix} 0.8167 & -0.2457 \\ -0.2457 & 1.2782 \end{bmatrix}
\end{aligned}
\tag{87}
$$

8.3 ML-MAS OFF-POLICY IRL USING NEURAL NETWORKS

The ML-MAS off-policy IRL Algorithm 2 is implemented by utilizing the collected data found by applying a fixed control policy u to the system to solve the Bellman equation for $V_i(\delta_i(t))$, and the updated control and disturbance policies u_i^{r+1} , u_j^{r+1} , and u_k^{r+1} iteratively. Three NNs, i.e., the actor NN, the critic NN, and the disturber NN, is used to approximate the cost function and the updated control and disturbance policies in the Bellman equation. The solution $V_i(\delta_i(t))$, and the updated control and disturbance policies is approximated by three NNs.

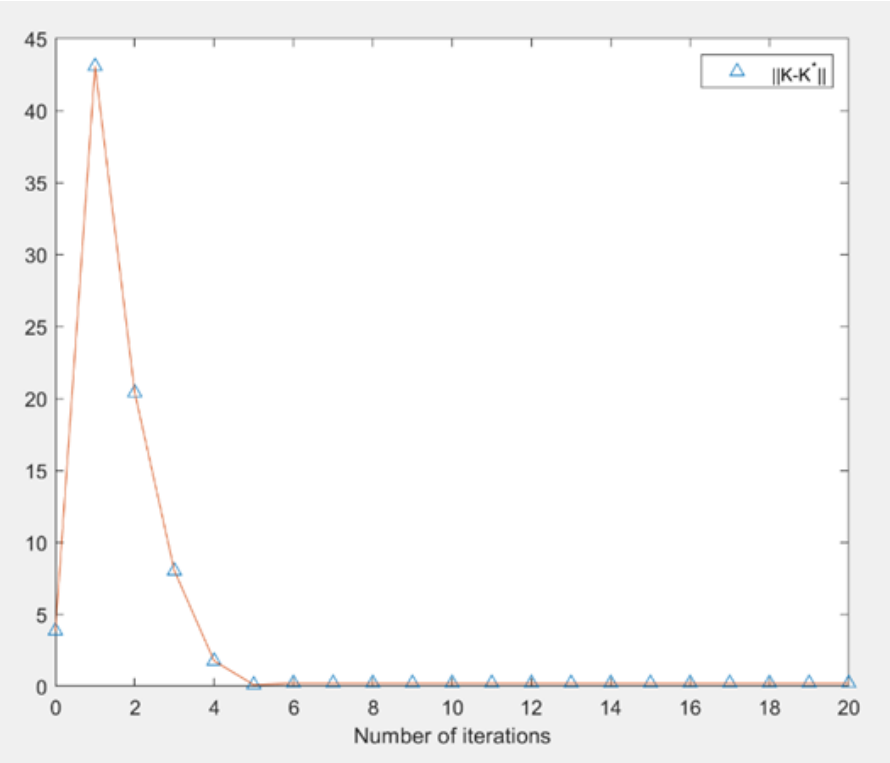


Figure 11 – ML-MAS Cluster Consensus – Convergence of the control gain to its optimal value

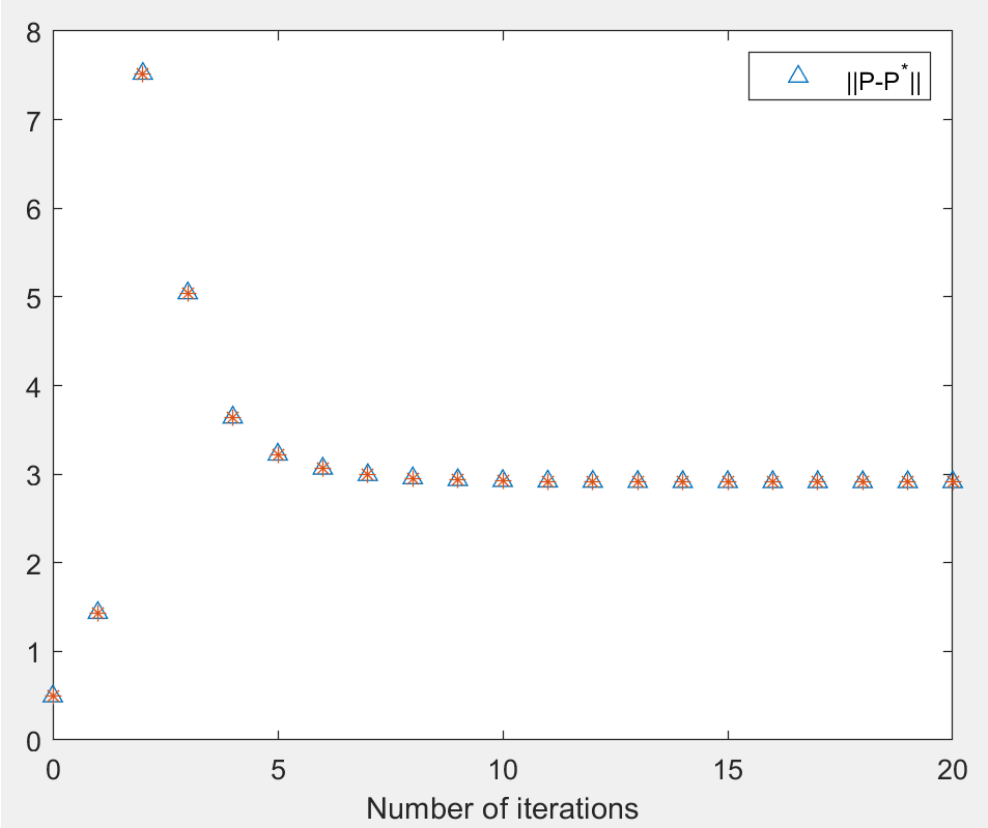


Figure 12 – ML-MAS Cluster Consensus – Convergence of the kernel matrix *P* to its optimal value

9 CONCLUSION

Multileader MASs cluster consensus has been proved under general topology with existing spanning tree, with NO limiting assumption on the communication graph. The underlying mechanisms between the system dynamics and the communication graph to reach cluster consensus are presented. The cluster consensus problem of Multileader MASs is investigated, where all clusters are allowed to have different communication weights. This extends existing results for clustering consensus from previous works.

An online off-policy reinforcement learning algorithm is developed to find the solution to the H∞ optimal problem of multileader MAS with completely unknown systems. This will extend the off-policy RL to Multileader MAS.

It is shown that, using off-policy RL, the disturbance input does not require to be specified and adjusted. Simulation results confirmed the suitability of the proposed method.

In this dissertation, the cluster consensus problem of Multileader has been considered. Chen has studied the heterogeneous MASs considering the heterogeneous dynamics and the negative couplings among agents [32]. Chen has restricted the communication topology between the cluster using zero-row sum assumption for Laplacian matrix to assure the cluster has no other Impact on other clusters. He designed the problem using Hamiltonian performance optimization. In this paper, the Chen's restrictions on the communication topology have been removed which allow the clusters to communicate with each other freely through the optimization. We have also used Min-Max differential game theory to optimize the state feedback control system. An important idea in this paper is to formulate the clustering problem of interconnected multileader systems as a disturbance attenuation problem as formulated in [5].

There have been several studies in the past to reach consensus in multiagent system. In this paper, for the first time, it has been proved that the multileader MAS can reach cluster consensus without limiting the communication between the clusters. The combination of Small Gain Theorem and H∞ optimization has been designed in the graphical differential game platform to prove the stability for the system.

After proving the cluster consensus for Multileader MASs, an online off-policy reinforcement learning algorithm is developed to find the solution to the H_∞ optimal problem of multileader MAS with completely unknown systems. The leaders and the agents of other clusters outside each agent

cluster will be defined as the system disturbance. It is not required that the disturbance be adjustable. An augmented system is constructed from the tracking error dynamics and the command generator dynamics for the H∞ optimal performance problem.

A performance HJI equation associated with the discounted performance function is derived, which gives both the feedforward and feedback parts of the control input simultaneously. An upper-bound and lower-bound is obtained for the discount factor to assure local asymptotic stability of the error dynamics using **Ultimately Uniformly Bounded (UUB)**. An off-policy RL algorithm is then developed to find the solution to the HJI equation online using only the measured data and without any knowledge about the system dynamics. Convergence of this algorithm to the solution to the HJI equation is shown.

The major contributions of this dissertation are as follows:

1. Multileader MASs cluster consensus has been proved under general topology with existing spanning tree, with NO limiting assumption on the communication graph. The underlying mechanisms between the system dynamics and the communication graph to reach cluster consensus are presented.
2. The cluster consensus problem of Multileader MASs is investigated, where all clusters are allowed to have different communication weights. This extends existing results in [32] for clustering consensus.
3. an online off-policy reinforcement learning algorithm is developed to find the solution to the $H\infty$ optimal problem of multileader MAS with completely unknown systems. This will extend the off-policy RL in [30] to Multileader MAS.

10 PUBLICATIONS

1. M. N. Naleini, A. T. Koru, Y. Kartal, V. G. Lopez and F. L. Lewis, "***Leader-Following Cluster Consensus as a Graphical Differential Game with a Nash Equilibrium Solution,***" in IEEE Control Systems Letters, vol. 6, pp. 2713-2718, 2022, doi: 10.1109/LCSYS.2022.3175665.

2. Naleini M. N., Koru A. T., and Lewis F. L. (2023): ***Leader-Following Consensus of Heterogeneous Uncertain Multi-agent Systems with a Distributed Adaptive Control Law***, International Journal of Adaptive Control and Signal Processing, 2017;00:1–6.

11 REFERENCES AND FOOTNOTES

11.1 REFERENCES

[1] H. L. F. D. A. Zhang, "Optimal design for synchronization of cooperative systems: state feedback, observer and output feedback," *IEEE Tranaction Automatic Control*, vol. 56, no. 8, pp. 1948-1952, 2011.

[2] K. W. J. Z. Y. e. a. Chen, "Second-order consensus of nonlinear multi-agent systems with restricted switching topology and time delay," *Nonlinear Dynamics,* vol. 78, no. 2, pp. 881-887, 2014.

[3] J. L. a. A. S. A. Jadbabaie, "Coordination of groups of mobile autonomous agents using nearest neighbor rules," *IEEE Transactions on Automatic Control,* vol. 48, no. 6, pp. 988 - 1001, 2003.

[4] T. B. a. P. Bernard, H∞-Optimal Control and Related Minimax Design Problems, Boston, MA, USA: Birkhasuser, 1995.

[5] D. V. a. V. L. S. F. L. Lewis, Optimal Control, New York, NY< USA: Wiley, 2012.

[6] M. D. S. Aliyu, Nonlinear H∞ Control, Hamiltonian Systems and Hamilton-Jacobi Equations, Boca Raton, FL, USA: CRC Press, 1996.

[7] P. K. A. J. K. J. A. Ball, "H∞ tracking control for a class of nonlinear systems," *IEEE Transactions Automatic Control,* vol. 44, no. 6, pp. 1202-1206, 1999.

[8] C. Papadimitriou and T. Roughgarden, "Computing correlated equilibria in multi-player games," *Journal of ACM,* vol. 55, no. 3, pp. 1-29, 2008.

[9] CALIFORNIA INST OF TECH PASADENA CONTROL AND DYNAMICAL SYSTEMS, "Flocking for multi-agent dynamic systems: Algorithms and theory," Defense Technical Information Center, 2004.

[10] M. E. P. S. S. Mercedes Delgado, "Defining clusters of related industries," *Journal of Economic Geography,* vol. 16, no. 1, pp. 1-38, 2015.

[11] B. J. Pierre Hansen, "Cluster analysis and mathematical programming," *Mathematical Programming,* vol. 79, no. 1-3, pp. 191-215, 1997.

[12] W. T. W. G. N. Lance, "A general theory of classificatory sorting strategies. i. hierarchical systems," *Computer Journal,* vol. 9, pp. 373-380, 1967.

[13] R. A. V. K. S. S. G. Kharypis, "Multilevel hypergraph partitioning: Applications in vlsi domain," in *Proceedings of the Design and Automation Conference,* 1997.

[14] R. C. Nam Nguyen, "Consensus Clusterings," in *IEEE ICDM,* Omaha, NE, USA, 2007.

[15] F. L. L. K. H. Movric, "Cooperative Optimal Control for Multi-Agent Systems on Directed Graph Topologies," *IEEE TRANSACTIONS ON AUTOMATIC CONTROL,* vol. 59, no. 3, pp. 769-775, 2014.

[16] W. L. T. C. Y. Han, "Achieving Cluster Consensus in Continuous-Time Networks of Multi-AgentsWith Inter-Cluster Non-Identical Inputs," *IEEE TRANSACTIONS ON AUTOMATIC CONTROL,* vol. 60, no. 3, pp. 793-798, 2015.

[17] C. Y. Jiahu Qin, "Cluster consensus control of generic linear multi-agent systems under directed topology with acyclic partition," *Automatica,* vol. 49, pp. 2898-2905, 2013.

[18] J. W. ,. Y. Z. F. L. L. Kairui Chen, "Cluster consensus of heterogeneous linear Multi-agent Systems," *IET Control Theory & Applications,* vol. 12, no. 11, pp. 1533-1542, 2018.

[19] J. Q. ,. H. G. C.B. Yu, "Cluster synchronization in directed networks of partial-state coupled linear systems under pinning control," *Automatica,* vol. 50, no. 9, pp. 2341-2349, 2014.

[20] C. Camerer, Behavioral Game Theory: Experiments in Strategic Interaction, Princton, NJ, USA: Princton University Press, 2003.

[21] J. Engwerda, LQ Dynamic Optimization and Differential Games, New York, NY, USA: Wiley, 2005.

[22] N. T. R. E. T. a. V. V. M. Kearns, "Graphical games," in *Algorithmic Game Theory*, vol. 3, Cambridge, U.K., Cambridge Univesity Press, 2007, pp. 159-180.

[23] L. Pavel, "A noncooperative game approach to OSNR optimization in optical networks," *IEEE Transactions Automatic Control,* vol. 51, no. 5, pp. 848-852, 2006.

[24] S. H. S. M. LaValle, "Optimal motion planning for multiple robots having independent goals," *IEEE Transactions Robotic Automatic,* vol. 14, no. 6, pp. 912-925, 1998.

[25] T. B. T. Alpcan, Network Security: A Decision and Game-Theoretic Approach, Cambridge, U.K.: Cambridge University Press, 2010.

[26] H. M. B. K. F. L. L. K. G. Vamvoudakis, "Game Theory-Based Contro System Algorithms with Real-time Reinforcement learning," *IEEE CONTROL SYSTEMS MAGAZINE,* vol. 1066, no. 033, pp. 33-42, 2017.

[27] A. G. B. R. S. Sutton, Reinforcement Learning: An Introduction, Cambridge, Ma, USA: MIT Press, 1998.

[28] D. V. K. G. V. F. L. Lewis, "Reinforcement learning and feedback control: Using natural decision methods to design optimal adaptive controllers," *IEEE Control System,* vol. 32, no. 6, pp. 76-105, 2012.

[29] K. G. V. F. L. L. D. Vrabie, Optimal Adaptive Control and Differential Games by Reinforcement Learning Principles (Control Engineering), Stevenage, U.K.: IET Press, 2012.

[30] F. L. L. Z.-P. J. Hamidreza Modares, "H∞ Tracking Control of Completely Unknown Continuous-Time Systems via Off-Policy Reinforcement Learning," *IEEE TRANSACTIONS ON NEURAL NETWORKS AND LEARNING SYSTEMS,* vol. 26, no. 10, pp. 2550-2562, 2015.

[31] F. L. L. H. Modares, "Linear quadratic tracking control of partially-unknown continuous-time systems using reinforcement learning," *IEEE Transaction Automatic Control,* vol. 59, no. 11, pp. 3051-3061, 2014.

[32] J. W. ,. Y. Z. F. L. L. Kairui Chen, "Cluster consensus of heterogeneous linear multi-agent systems," *IET Control Theory & Applications,* vol. 12, no. 11, pp. 1533-1542, 2018.

[33] P. B. T. Ba¸sar, H∞-Optimal Control and Related Minimax Design Problems, Boston, MA, USA: Birkhäuser, 1995.

[34] F. L. L. Z.-P. J. Hamidreza Modares, "H∞ Tracking Control of Completely Unknown Continuous-Time Systems via Off-Policy Reinforcement Learning," *IEEE TRANSACTIONS ON NEURAL NETWORKS AND LEARNING SYSTEMS,* vol. 26, no. 10, pp. 2550-2562, 2015.

[35] C. Yue, E. Thomas and M. Vasilios, "On infinite-time nonlinear quadratic optimal control," *System & Control Letters,* vol. 51, no. 3-4, pp. 259-268, 2004.

[36] A. J. V. d. Schaft, "L2-gain analysis of nonlinear systems and nonlinear state-feedback H∞ control," *IEEE Transactions Automatic Control,* vol. 37, no. 6, pp. 770-784, 1992.

[37] M. J. L. G. Corless, "Continuous State Feedback guaranteeing Uniform Ultimate Boundedness for Uncertain Dynamic Systems," *IEEE transactions on Automatic Control,* Vols. AC-26, no. 5, pp. 1139-1144, 1981.

[38] Z. D. Z. C. G. Li, "On H∞ and H2 performance regions of multiagent system," *Automatica,* vol. 47, no. 4, p. 797–803, 2011.

[39] H. K. Khalil, Nonlinear Systems, NJ, USA: Princton-Hall, 2002.

[40] K. T. N. M. Harry L. Trentelman, "Robust Synchronization of Uncertain Linear Multi-Agent Systems," *IEEE TRANSACTIONS ON AUTOMATIC CONTROL,* vol. 58, no. 6, pp. 1511-1523, 2013.

[41] H. Zhang and F. L. Lewis, "Adaptive cooperative tracking control of higher-order nonlinear systems with unknown dynamics," *Automatica,* vol. 48, no. 7, pp. 1432-1439, 2012.

[42] S. Lyshevski, "Optimal Control of Nonlinear Continuous-Time Systems: Design of Bounded Controllers via Generalized Nonquadratic Functionals," *Proc. American Control Cenference,* pp. 205-209, 1998.

[43] F. L. Draguna Vrabie, "Neural network approach to continuous-time direct adaptive optimal control for partially unknown nonlinear systems," *Neural Networks,* no. 22, pp. 237-246, 2009.

[44] B. L. H.-N.Wu, "Neural network based online simultaneous policy update algorithm for solving the HJI equation in nonlinear H∞ control," *IEEE Transactions Neural Network Learning Systems,* vol. 23, no. 12, pp. 1884-1895, 2012.

[45] Z.-P. J. Y. Jiang, "Computational adaptive optimal control for continuous-time linear systems with completely unknown dynamics," *Automatica,* vol. 48, no. 10, pp. 2699-2704, 2012.

[46] B. Kiumarsi and F. L. Lewis, "Actor–critic-based optimal tracking for partially unknown nonlinear discrete-time systems," *IEEE Transaction Neural Networks Learning Systems,* vol. 1, no. 26, pp. 140-151, 2015.

[47] F. L. L. H. M. A. K. M. B. N.-S. B. Kiumarsi, "Reinforcement Q-learning for optimal tracking control of linear discrete-time systems with unknown dynamics," *Automatica,* vol. 50, no. 4, p. 1167–1175, 2014.

[48] R. S. S. a. A. G. Barto, Reinforcement Learning: An Introduction, Cambridge, MA: MIT Press, 1998.

[49] H. L. F. D. A. Zhang, "Optimal Design for Synchronization of Cooperative Systems: State Feedback, bserver and Output Feedback," *IEEE Transactions on Automatic Control,* vol. 56, pp. 1948-1952, 2011.

[50] S. M. A. Lyshevski, "Control System Analysis and Design Upon the Lyapunav Method," *Proc. American Control Conference,* pp. 3219-3223, 1995.

11.2 **FOOTNOTES**