Deep and Decentralized Multi-Agent Coverage of a Target with Unknown Distribution

Hossein Rastgoftar

***Abstract*—This paper proposes a new architecture for multi-agent systems to cover an unknown distributed target quickly and safely and in a decentralized manner. The inter-agent communication is organized by a directed graph with a fixed topology and We model agent coordination as a decentralized leader-follower problem with time-varying communication weights. Given this problem setting, we first present a method for converting the communication graph into a neural network, where an agent can be represented by a unique node of the communication graph but multiple neurons of the corresponding neural network. We then apply a mass-centric strategy to train time-varying communication weights of the neural network in a decentralized fashion. This implies that the observation zone of every follower agent is independently assigned by the follower based on positions of its in-neighbors. By training the neural network, we can ensure safe and decentralized multi-agent coverage control. Despite the target is unknown to the agent team, we provide a proof for convergence of the proposed multi-agent coverage method. The functionality of the proposed method is validated by a large-scale multi-copter team covering distributed targets on the ground.**

***Index Terms*—Large-Scale Coordination, Multi-Agent Coverage, and Decentralized Control.**

I. Introduction

Multi-agent coverage has received a lot of attention from the control community over the past few years. Multi-agent coverage has many applications, such as wildfire management [1], [2], border security [3], agriculture [4], [5], and wildlife monitoring [6]. A variety of coverage approaches have been proposed by the researchers that are reviewed in Section I-A.

A. Related Work

Sweep [7], [8] and Spiral [9], [10] are two available methods used for single-vehicle coverage path planning, while Vehicle Routing Problem [11], [12] is widely used for multi-agent coverage path planning. Stability and convergence of diffusion-based multi-agent coverage are studied in Ref. [13]. Decentralized multi-agent coverage using local density feedback is achieved by applying the discrete-time mean-field model in Ref. [14]. Multi-agent coverage conducted by unicycle robots, that is guided by a single leader, is investigated in Ref. [15], where the authors propose to decouple coordination and coverage modes. Adaptive decentralized multi-agent coverage is studied in [16], [17]. Ref. [18] offers a multiscale analysis of multi-agent coverage control that provides the convergence properties in continuous time. Human-centered active sensing of wildfire by unmanned aerial vehicles is studied in Ref. [2]. Ref. [19] suggests applying the k-means algorithm for the planning of zone coverage by multiple agents. Reinforcement Learning-(RL-)-based multi-agent coverage control is investigated by Refs. [4], [20]–[23].

H. Rastgoftar is with the Department of Aerospace and Mechanical Engineering, University of Arizona, Tucson, AZ, 85721 USA e-mail: hrastgoftar@arizona.edu.

Authors in [20], [24]–[27] used the Vononoi-based approach for covering a distributed target. This widely accepted approach considers multi-agent coverage by N agents, defined by set $\mathcal{V} = \{1, \cdots, N\}$, and decomposes search (coverage) domain $C \subset \mathbf{R}^n$ by defining Voronoi partition

$$V_i(\mathbf{q}) = \left\{ \mathbf{q} \in C : \|\mathbf{q} - \mathbf{r}_i\|^2 \leq \|\mathbf{q} - \mathbf{r}_j\|^2, \ \forall j \in \mathcal{V} \setminus \{i\} \right\}, \ \forall i \in \mathcal{V}, \tag{1}$$

where $\mathbf{r}_i \in C$ is the position of agent $i \in \mathcal{V}$. The coverage cost is then defined by

$$H(\mathbf{q}) = \frac{1}{2} \sum_{i \in \mathcal{V}} \int_{V_i(\mathbf{q})} \|\mathbf{q} - \mathbf{r}_i\| \rho(\mathbf{q}) \, d\mathbf{q}, \tag{2}$$

where the proof for multi-agent coverage convergence is available. Note that most of the available work uses the Lyapunov Theorem to provide the proof for the convergence guarantee, where they model every agent $i \in \mathcal{V}$ by a single integrator dynamics. Voronoi-based coverage in the presence of obstacles and failures is presented as a leader-follower problem in Ref. [24]. Ref. [28] experimentally evaluates the functionality of the Voronoi-based and other multi-agent coverage approaches in urban environments.

B. Contributions

Voronoi-based control [20], [24]–[27] is a main approach for multi-agent coverage widely studied by the control community. Voroni-based cell decomposition requires positions of all agents to assign the Voroni-based zone allocated to individual agents. As a result, achieving a truly decentralized coverage may be difficult or at least computationally expensive, if possible. This paper develops a method for decentralized multi-agent coverage of a distributed target with an unknown distribution that is specified by a discrete set. We propose to define inter-agent communications by a deep neural network (DNN), which is called *coverage neural network*, with time-varying weights that are obtained such that coverage convergence is ensured. To this end, the paper establishes specific rules for structuring the coverage neural network and proposes a mass-centric approach to train the DNN weights at any time t that specify inter-agent communication among the agent team. Although the target is unknown to the agent team, we prove that the weights ultimately converge to unique values that quantify target distribution in the motion space. The functionality of the proposed coverage method will be validated by simulating aerial coverage conducted by a team of quadcopter agents. Compared to the existing work, this paper offers the following novel contributions:

Contribution 1: The paper develops a method for decentralized partitioning and coverage of an unknown distributed target, with discrete positions in $\mathbb{R}^n$, where $n = 2,3$ is the dimension of the motion space. Compared to the Voronoi-based approach [20], [24]–[27], the proposed DNN-based multi-agent coverage does not apply the Voronoi-based partitioning by using Eq. (1), **based on the current positions of the agents**. Indeed, the DNN structure, specifying inter-agent communications, is fixed and assigned based on the agents' initial positions.

Contribution 2: The proposed multi-agent coverage approach learns the inter-agent communication weights in a forward manner as opposed to the existing neural network learning method, which trains weights by combining forward and backward iterations [29]. To be more precise, every neuron represents an agent, with the output specifying the agent's actual position, where the DNN weights of every non-input layer are solely assigned based on the outputs of the connected neurons of the previous layer in real time. More specifically, weights input to a hidden layer are assigned based on the (i) outputs of the previous layer and (i) target data information independently measured by observing the neighboring environment. We provide proof of convergence for the proposed learning approach.

Contribution 3: We propose an algorithmic approach for structuring inter-agent communications based on the agent team initial formation arbitrarily distributed in $\mathbb{R}^n$ such that the communication graph can be converted to a DNN. The proposed approach classifies agents as leaders and followers and ensures that every follower agent communicates with $n+1$ in-neighbors, where the in-neighbor agents form a n-D simplex enclosing the follower. The DNN structure remains time-invariant but the DNN weights are time-varying.

C. Outline

The remainder of the paper is organized as follows: The problem statement and formulation are given in Section II. The paper methodology is presented in Section III. Assuming every agent is a quadcopter, the multi-agent network dynamics are obtained in Section IV, followed by simulation results in Section V, and a conclusion in Section VII.

II. Problem Statement and Formulation

We consider a team of N agents identified by set $\mathcal{V} = \{1,\cdots,N\}$ and classify them into the following three groups:

1) "boundary" agents identified by $\mathcal{V}_B = \{b_1,\cdots,b_{N_B}\}$ are distributed along the boundary of the agent team configuration;
2) a single "core" agent identified by singleton $\mathcal{V}_C$ is an interior agent; and
3) follower agents defined by $\mathcal{V}_I = \mathcal{V}\setminus(\mathcal{V}_B\bigcup\mathcal{V}_C)$ are all located inside the agent team configuration.

Note that $\mathcal{V}$ can be expressed as $\mathcal{V} = \mathcal{V}_B\bigcup\mathcal{V}_C\bigcup\mathcal{V}_I$, where $\mathcal{V}_B$, $\mathcal{V}_C$, and $\mathcal{V}_I$ are disjoint subsets of $\mathcal{V}$, i.e. $\mathcal{V}_B\bigcap\mathcal{V}_C = \emptyset$, $\mathcal{V}_C\bigcap\mathcal{V}_I = \emptyset$, and $\mathcal{V}_I\bigcap\mathcal{V}_B = \emptyset$. Inter-agent communication among the agents are defined by graph $\mathcal{G}(\mathcal{V},\mathcal{E})$ where $\mathcal{E}\subset\mathcal{V}\times\mathcal{V}$ defines edges of graph $\mathcal{G}$ and each edge represents a unique communication link. More specifically, if $(j,i)\in\mathcal{E}$, then i accesses the position of $j\in\mathcal{V}$.

Definition 1. We define

$$\mathcal{N}(i) = \{j\in\mathcal{V} : (j,i)\in\mathcal{E}\},\qquad \forall i\in\mathcal{V}. \tag{3}$$

as the set of in-neighbors of agent $i\in\mathcal{V}$.

A. Neural Network Representation of Inter-Agent Communication

The reference (initial) configuration of the agents is used to specify the graph $\mathcal{G}(\mathcal{V},\mathcal{E})$, which can be represented by a deep neural network with $M+1$ layers. We use set $\mathcal{M} = \{0,\cdots,M\}$ to define the layer identification numbers. Therefore, set $\mathcal{V}$ can be expressed as

$$\mathcal{V} = \bigcup_{l\in\mathcal{M}}\mathcal{V}_l, \tag{4}$$

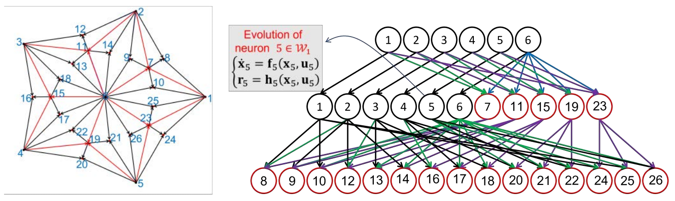
where $\mathcal{V}_0$ through $\mathcal{V}_M$ are disjoint subsets of $\mathcal{V}$. We use $\mathcal{W}_0$, $\mathcal{W}_1$, $\cdots$, $\mathcal{W}_M$ to identify the neurons of layers 0 through M of the coverage neural network, and $\mathcal{W}_l$ and $\mathcal{V}_l$ are related by

$$\mathcal{W}_l = \begin{cases}\mathcal{V}_l & l\in\{0,M\}\\ \mathcal{W}_{l-1}\bigcup\mathcal{V}_l & l\in\mathcal{M}\setminus\{0,M\}\end{cases}. \tag{5}$$

Note that $\mathcal{W}_0 = \mathcal{V}_0 = \mathcal{V}_B\bigcup\mathcal{V}_C$ defines neurons that uniquely represent boundary and core agents. Also, $\mathcal{V}_0 = \mathcal{V}_B\bigcup\mathcal{V}_C$ defines the leader agents and $\mathcal{V}\setminus\mathcal{V}_0$ defines the follower agents. $\mathcal{N}(i) = \emptyset$, if $i\in\mathcal{V}_0$ is a leader.

Definition 2. For every neuron $i\in\mathcal{W}_l$ at layer $l\in\mathcal{M}\setminus\{0\}$, $\mathcal{I}(i,l)\in\mathcal{W}_{l-1}$ defines those neurons of $\mathcal{W}_{l-1}$ that are connected to $i\in\mathcal{W}_l$. Assuming the agent team forms an n-dimensional configuration in a three-dimensional motion space ($n = 2,3$), we use the following key rules to define $\mathcal{I}(i,l)$ for every $i\in\mathcal{W}_l$ and $l\in\mathcal{M}\setminus\{0\}$:

$$|\mathcal{I}(i,l)| = \begin{cases}1 & \text{If } i\in\mathcal{W}_{l-1}\bigcap\mathcal{W}_l \text{ and } l\in\mathcal{M}\setminus\{0\}\\ n+1 & \text{If } i\in\mathcal{W}_l-\mathcal{W}_{l-1} \text{ and } l\in\mathcal{M}\setminus\{0\}\\ n+1 & \text{If } i\in\mathcal{W}_M\\ 0 & \text{If } i\in\mathcal{W}_0\end{cases}. \tag{6}$$


Evolution of neuron 5 ∈ $\mathcal{W}_1$
$\begin{cases}\dot{\mathbf{x}}_5 = \mathbf{f}_5(\mathbf{x}_5,\mathbf{u}_5)\\ \mathbf{r}_5 = \mathbf{h}_5(\mathbf{x}_5,\mathbf{u}_5)\end{cases}$

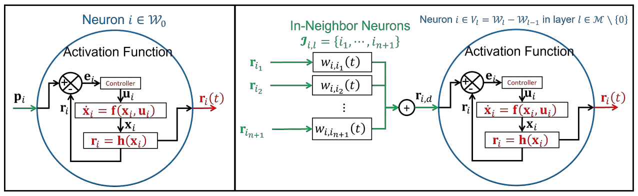
(a) (b)

Fig. 1: (a) Graph $\mathcal{G}(\mathcal{V},\mathcal{E})$ representing the coverage neural network that consists of three layers ($\mathcal{M} = \{0,1,2\}$), where $\mathcal{V} = \mathcal{V}_0\bigcup\mathcal{V}_1\bigcup\mathcal{V}_2$, $\mathcal{V} = \{1,\cdots,26\}$, $\mathcal{V}_0 = \{1,2,\cdots,6\}$, $\mathcal{V}_1 = \{7,11,15,19,23\}$, and $\mathcal{V}_2 = \{8,9,10,12,13,14,16,17,18,20,21,22,24,25,26\}$. (b) Neural network representation of the communication graph $\mathcal{G}$.

We note that $\mathcal{N}(i)$ and $\mathcal{I}(i,l)$ can be related by

$$\bigwedge_{l\in\mathcal{M}\setminus\{0\}}\ \bigwedge_{i\in\mathcal{W}_l\setminus\mathcal{W}_{l-1}}\left(\mathcal{I}(i,l) = \mathcal{N}(i)\right). \tag{7}$$

For better clarification, we consider an agent team with $N = 26$ agents identified by set $\mathcal{V} = \{1,\cdots,26\}$, where the team forms a two-dimensional configuration ($n = 2$) shown in Fig. 3 (a). The inter-agent communications shown in Fig. 3 (a) can be represented by the neural network of Fig. 3 (b) with three layers $\mathcal{M} = \{0, 1, 2\}$. In this example, $\mathcal{W}_0 = \{1, \cdots, 6\}$, defining the boundary and core leaders, has no in-neighbors, and $\mathcal{W}_2 = \{8, 9, 10, 12, 13, 14, 16, 17, 18, 20, 21, 22, 24, 25, 26\}$, defining followers, each has three in-neighbors. Also, $\{7, 11, 15, 19, 23\} \in \mathcal{W}_1$ each has three in-neighbors, but the remaining neurons of $\{1, \cdots, 6\} \in \mathcal{W}_1$ that are repeated in layer 0 each have one in-neighbor.


Neuron $i \in \mathcal{W}_0$
Activation Function
$\mathbf{p}_i$
$\mathbf{e}_i$
Controller
$\mathbf{u}_i$
$\mathbf{r}_i$
$\dot{\mathbf{x}}_i = \mathbf{f}(\mathbf{x}_i, \mathbf{u}_i)$
$\mathbf{x}_i$
$\mathbf{r}_i = \mathbf{h}(\mathbf{x}_i)$
$\mathbf{r}_i(t)$
In-Neighbor Neurons
$\mathcal{I}_{i,l} = \{i_1, \cdots, i_{n+1}\}$
Neuron $i \in V_l = \mathcal{W}_l - \mathcal{W}_{l-1}$ in layer $l \in \mathcal{M} \setminus \{0\}$
$\mathbf{r}_{i_1}$
$\mathbf{r}_{i_2}$
$\mathbf{r}_{i_{n+1}}$
$w_{i,i_1}(t)$
$w_{i,i_2}(t)$
$w_{i,i_{n+1}}(t)$
$\mathbf{r}_{i,d}$
Activation Function
$\mathbf{e}_i$
Controller
$\mathbf{u}_i$
$\mathbf{r}_i$
$\dot{\mathbf{x}}_i = \mathbf{f}(\mathbf{x}_i, \mathbf{u}_i)$
$\mathbf{x}_i$
$\mathbf{r}_i = \mathbf{h}(\mathbf{x}_i)$
$\mathbf{r}_i(t)$
Fig. 2: Activation functions, inputs and outputs of neurons of the coverage neural network. (Left): An example neuron in $i \in \mathcal{W}_0$. (Right): An example neuron $i \in$ in layer $l \in \mathcal{M} \setminus \{0\}$

B. Differential Activation Function

The DNN neurons represent agents, where each neuron receives the agent's desired position as the input and returns the agent's actual position as the output. Therefore, unlike the available neural network, neurons of the coverage neural network are operated by differential activation functions given by

$$\begin{cases} \dot{\mathbf{x}}_i = \mathbf{f}_i(\mathbf{x}_i, \mathbf{u}_i) \\ \mathbf{r}_i = \mathbf{h}_i(\mathbf{x}_i) \end{cases}, \qquad i \in \mathcal{W}_l, \ l \in \mathcal{M}. \tag{8}$$

In Eq. (8), $\mathbf{x}_i \in \mathbb{R}^{n_{x,i}}$ and $\mathbf{u}_i \in \mathbb{R}^{n_{u,i}}$ denote the state vector and the control of neuron i, respectively, and $\mathbf{h}_i : \mathbb{R}^{n_{x,i}} \to \mathbb{R}^3$, $\mathbf{f}_i : \mathbb{R}^{n_{x,i}} \to \mathbb{R}^{n_{x,i}}$, and $\mathbf{g}_i : \mathbb{R}^{n_{x,i}} \to \mathbb{R}^{n_{x,i} \times n_{u,i}}$ are smooth functions (See Fig. 2).

Note that neuron $i \in \mathcal{W}_l$ represents agent $i \in \mathcal{V}$ where the the nonlinear differential activation function, given by Eq. (8), is indeed the dynamics of agent $i \in \mathcal{V}$. Therefore, the output of neuron i, denoted by $\mathbf{r}_i \in \mathbb{R}^{3\times1}$ is the actual position of agent i at time t. The input of neuron i is indeed the desired trajectory of agent $i \in \mathcal{V}$ and obtained by

$$\mathbf{r}_{i,d}(t) = \begin{cases} \mathbf{p}_i \quad \text{(given)} & i \in \mathcal{W}_0 \\ \sum_{j \in \mathcal{I}(i,l)} w_{ij}(t)\mathbf{r}_j(t) & i \in \mathcal{W}_l, \ l \in \mathcal{M} \setminus \{0\} \end{cases}, \tag{9}$$

where $\mathbf{p}_i$ is a desired constant position for leader agent $i \in \mathcal{V}_0$ ($\mathcal{V}_0 = \mathcal{W}_0$). Also, $w_{i,j}(t) > 0$ is the time-varying communication weight between $i \in \mathcal{W}_l \setminus \mathcal{W}_{l-1}$ and $j \in \mathcal{I}(i,l)$, and satisfies the following constraint:

$$\bigwedge_{l \in \mathcal{M} \setminus \{0\}} \bigwedge_{i \in \mathcal{W}_l \setminus \mathcal{W}_{l-1}} \left(\sum_{j \in \mathcal{I}(i,l)} w_{i,j}(t) = 1 \right), \qquad \forall t. \tag{10}$$

C. Objectives

Given above problem setting, this paper offers a DNN-based method for optimal coverage of target set $\mathcal{D}$ with distribution that is unknown to the agent team. To achieve this objective, we assume that positions of boundary and core leaders, defined by $\mathcal{V}_0$, are known, and solve the following three main problems:

1) **Problem 1–Establishing Inter-Agent Communication:** The communication graph $\mathcal{G}(\mathcal{V}, \mathcal{E})$ needs to be structured such that it can be converted to the a neural network with the properties specified above. We propose an algorithmic approach to obtain graph $\mathcal{G}(\mathcal{V}, \mathcal{E})$ based on the initial configuration of the agent team.
2) **Problem 2–Abstract Representation of Target:** We develop a mass-centric approach in Section III-B to abstractly represent target by position vectors $\mathbf{p}_1$ through $\mathbf{p}_N$ that are considered as agents' final desired positions.
3) **Problem 3–Decentralized Target Acquisition:** We propose a forward method to train the the DNN weights, and assign control input $\mathbf{u}_i$ such that actual position $\mathbf{r}_i$ converges to the desired position $\mathbf{p}_i$ in a decentralized fashion, for every $i \in \mathcal{V} \setminus \mathcal{W}_0$. By decentralized convergence, we imply that agent $i \in \mathcal{V} \setminus \mathcal{W}_0$ does <u>**not**</u> know global position $\mathbf{p}_i$ but actual position $\mathbf{r}_i(t)$ asymptotically converges to $\mathbf{p}_i$, as $t \to \infty$.

III. Methodology

We consider an agent team that is contained by a convex polytope with vertices that are occupied by N_B boundary leader agents b_1 through b_{N_B}. This convex polytope, defined by the boundary agents, is called *leading polytope*. The agent team applies the DNN-based coverage method to cover a distributed target that is specified by finite and discrete set $\mathcal{D} = \{1, \cdots, n_d\}$, where $\mathbf{d}_i \in \mathbb{R}^{3\times1}$ is the position of target data $i \in \mathcal{D}$. Set $\mathcal{D}$ defines n_d entities, where agents can use their onboard sensors to identify each target data (entity). For example, in search and rescue operations, the target set $\mathcal{D}$ can represent n_d persons affected by a tragedy on the ground, which can be pinpointed by heat from the drones' onboard sensors. The agents team can be made up of N multicopter drones.

Assumption 1. *The target set $\mathcal{D}$ is contained by the leading polytope with vertices that are known and considered as desired positions of boundary agents. However, the target distribution inside the leading polytope is unknown to the agent team.*

Assumption 2. *Boundary leader agents form an $n-D$ polytope in $\mathbb{R}^n$, thus, the boundary agents' desired positions must satisfy the following rank condition:*

$$\text{rank}\left(\begin{bmatrix} \mathbf{p}_{b_2} - \mathbf{p}_{b_1} & \cdots & \mathbf{p}_{b_{N_B}} - \mathbf{p}_{b_1} \end{bmatrix}\right) = n \tag{11}$$

If Assumption 2 is satisfied, the leading polytope, defined by the boundary agents, can be decomposed into N_L disjoint n-dimensional simplexes all sharing the core agent, defined by $\mathcal{V}_C$. We use $c_{b_1, b_2, \cdots, b_{N_B}} \in \mathcal{V}$ to identify the core agent, i.e. $\mathcal{V}_C = \left\{c_{b_1, b_2, \cdots, b_{N_B}}\right\}$. These leading simplex cells are identified by set $\mathcal{L} = \{1, \cdots, N_L\}$, where $\mathcal{S}_i = \left\{h_{i,1}, \cdots, h_{i,n}, c_{b_1, b_2, \cdots, b_{N_B}}\right\}$ defines vertices of simplex cell $i \in \mathcal{L}$ and

$$\bigwedge_{i \in \mathcal{L}} \left(\text{rank}\left(\begin{bmatrix} \mathbf{p}_{h_{i,1}} - \mathbf{p}_{N_B+1} & \cdots & \mathbf{p}_{h_{i,n}} - \mathbf{p}_{N_B+1} \end{bmatrix}\right) = n\right). \tag{12}$$

Remark 1. Note that $h_{i,1},\cdots,h_{n,i}\in\mathcal{S}_i\setminus\mathcal{V}_C$ are the boundary agents which in turn implies that

$$\bigwedge_{i\in\mathcal{L}}\left((\mathcal{S}_i\setminus\mathcal{V}_C)\subset\mathcal{V}_B\right). \tag{13}$$

Also, $\mathcal{W}_0=\mathcal{V}_0$ can be expressed as follows:

$$\mathcal{W}_0=\bigcup_{i\in\mathcal{L}}\mathcal{S}_i. \tag{14}$$

For better clarification, the 2-D ($n=2$) agent team configuration shown in Fig. 1 is a pentagon with vertices occupied by boundary agents $b_1=1$, $b_2=2$, $b_3=3$, $b_4=4$, and $b_5=5$. The pentagon can be decomposed into $N_L=5$ triangles, all sharing the agent $c_{1,2,3,4,5}=6$. Vertices of these triangular cells are defined by $\mathcal{S}_1=\{1,2,6\}$, $\mathcal{S}_2=\{2,3,6\}$, $\mathcal{S}_3=\{3,4,6\}$, $\mathcal{S}_4=\{4,5,6\}$, and $\mathcal{S}_5=\{5,1,6\}$.

Assumption 3. *Every agent $i\in\mathcal{V}\setminus\mathcal{V}_0$ has $n+1$ in-neighbors, therefore,*

$$\bigwedge_{l\in\mathcal{M}\setminus\{0\}}\bigwedge_{i\in\mathcal{V}_l}\left(|\mathcal{N}(i)|=n+1\right). \tag{15}$$

Assumption 3 implies that the DNN structure holds the following properties:

$$\bigwedge_{l\in\mathcal{M}\setminus\{0\}}\bigwedge_{i\in\mathcal{W}_l\setminus\mathcal{W}_{l-1}}\left(|\mathcal{I}(i,l)|=n+1\right), \tag{16a}$$

$$\bigwedge_{l\in\mathcal{M}\setminus\{0\}}\bigwedge_{i\in\mathcal{W}_l\bigcap\mathcal{W}_{l-1}}\left(|\mathcal{I}(i,l)|=1\right). \tag{16b}$$

In this paper, we use $\mathcal{V}$, $\mathcal{V}_l$, and $\mathcal{N}(i)$ when we refer to actual agents. On the other hand, we use $\mathcal{W}$, $\mathcal{W}_l$, and $\mathcal{I}(i,l)$ when we refer to the neurons representing actual agents.

Assumption 4. *The in-neighbors of every agent $i\in\mathcal{V}\setminus\mathcal{W}_0$ defined by $\mathcal{N}(i)=\{j_1,\cdots,j_{n+1}\}$ forms an n-D simplex. This condition can be formally specified as follows:*

$$\bigwedge_{l\in\mathcal{M}\setminus\{0\}}\bigwedge_{i\in\mathcal{V}_l}\left(\text{rank}\left(\begin{bmatrix}\mathbf{p}_{j_2}-\mathbf{p}_{j_1} & \cdots & \mathbf{p}_{j_{n+1}}-\mathbf{p}_{j_1}\end{bmatrix}\right)=n\right). \tag{17}$$

Indeed Assumption 3 is a necessary condition for Assumption 4. If Assumptions 3 and 4 are satisfied, the desired position of an agent $i\in\mathcal{V}_l$, for $l\in\mathcal{M}\setminus\{0\}$, can be uniquely expressed as a weighted sum of its in-neighbors, defined by $\mathcal{N}(i)=\mathcal{I}(i,l)$, where the weights are all unique at any time t. This property will be used in Section III-C to obtain the training weights of the DNN in a forward manner.

Definition 3. For every agent $i\in\mathcal{V}\setminus\mathcal{W}_0$,

$$\bar{C}_i=\left\{\sum_{j\in\mathcal{I}(i,l)}\sigma_j\mathbf{p}_j:\sigma_j\geq 0\text{ and }\sum_{j\in\mathcal{I}(i,l)}\sigma_j=1\right\},\ i\in,\ l\in\mathcal{M}, \tag{18a}$$

$$C_i(t)=\left\{\sum_{j\in\mathcal{I}(i,l)}\sigma_j\mathbf{r}_j(t):\sigma_j\geq 0\text{ and }\sum_{j\in\mathcal{I}(i,l)}\sigma_j=1\right\},\ i\in,\ l\in\mathcal{M}, \tag{18b}$$

define the convex hulls specified by "desired" and "actual" positions of agent i's in-neighbors, respectively.

Definition 4. We define

$$C=\bigcup_{l\in\mathcal{M}\setminus\{0\}}\bigcup_{i\in\mathcal{W}_l\setminus\mathcal{W}_{l-1}}\bar{C}_i \tag{19a}$$

$$C=\bigcup_{l\in\mathcal{M}\setminus\{0\}}\bigcup_{i\in\mathcal{W}_l\setminus\mathcal{W}_{l-1}}C_i(t) \tag{19b}$$

as the coverage zone that encloses all data points defined by set $\mathcal{D}$.

By considering Definition 3, we can express set $\mathcal{D}$ as

$$\mathcal{D}=\bigcup_{i\in\mathcal{I}(i,l)}\bar{\mathcal{D}}_i\qquad\text{or}\qquad\mathcal{D}=\bigcup_{i\in\mathcal{I}(i,l)}\mathcal{D}_i(t),$$

where

$$\bar{\mathcal{D}}_i=\left\{j\in\mathcal{D}:\mathbf{d}_j\in\bar{C}_i\right\} \tag{20}$$

is the target set that is "desired" to be searched by follower agent $i\in\mathcal{V}\setminus\mathcal{V}_0$ whereas

$$\mathcal{D}_i(t)=\left\{j\in\mathcal{D}:\mathbf{d}_j\in C_i(t)\right\} \tag{21}$$

is a subset of $\mathcal{D}$ that is "actually" searched by follower agent $i\in\mathcal{V}\setminus\mathcal{V}_0$ at time t. Note that $\bar{\mathcal{D}}_i$ and $\mathcal{D}_i(t)$ are enclosed by the convex hulls $\bar{C}_i$ and $C_i(t)$, respectively.

Assumption 5. *We assume that $\bar{\mathcal{D}}_i\neq\emptyset$ and $\mathcal{D}_i(t)\neq\emptyset$, at any time t, for every $i\in\mathcal{V}\setminus\mathcal{W}_0$.*

Assumption 5 implies that every simplex C_i and $\bar{C}_i$ must at least contain one target data. This assumption is indeed required to archive multi-agent coverage of an unknown distributed target in a truly decentralized fashion.

Algorithm 1 Structuring DNN based on initial formation

```
1: Get: Initial (reference) position of every agent $i\in\mathcal{V}$.
2: Outcome: Structuring Graph $\mathcal{G}(\mathcal{V},\mathcal{E})$ such that it can be
   converted to a deep neural network.
3: Assign boundary agents $\mathcal{V}_B$.
4: Assign core leader agent $c(\mathcal{V}_B)$.
5: Specify leading simplexes $\mathcal{S}_1$ through $\mathcal{S}_{N_L}$.
6: Define $O=\mathcal{S}_1\cup\cdots\cup\mathcal{S}_{N_L}$.
7: $check=0$ and $l=0$.
8: while $check=0$ do
9:     $l\leftarrow l+1$.
10:    $\mathcal{V}_l=\emptyset$.
11:    $\mathcal{X}=\emptyset$.
12:    for Every $\mathcal{H}$ in $O$ do
13:        Specify $CONT(\mathcal{H})$ using Definition 6.
14:        if $CONT(\mathcal{H})\neq\emptyset$ then
15:            Assign mentee agent $c(\mathcal{H})$ using Definition 7.
16:            Define in-neighbors of $c(\mathcal{H})$: $\mathcal{N}(c(\mathcal{H}))=\mathcal{H}$.
17:            $\mathcal{X}\leftarrow\mathcal{X}\bigcup\mathcal{H}$.
18:            $\mathcal{V}_l\leftarrow\mathcal{V}_l\bigcup\{c(\mathcal{H})\}$.
19:        end if
20:    end for
21:    if $\mathcal{X}=\emptyset$ then
22:        $check=1$.
23:        $M=l-1$.
24:    else
25:        $O\leftarrow\mathcal{X}$.
26:    end if
27: end while
28:
```

A. Inter-Agent Communication Structure

Inter-agent communication structure, defined by a DNN, is obtained based on the reference (initial) configuration of the agent team where agent $i\in\mathcal{V}$ is positioned at $\mathbf{a}_i\in\mathbb{R}^n$. To structure the DNN, we propose Algorithm 1 to divide agents into $M+1$ groups, defined by $\mathcal{V}_0$ through $\mathcal{V}_M$, and obtain in-neighbor set $\mathcal{N}(i)\subset\mathcal{W}_{l-1}$, for every $l\in\mathcal{M}\setminus\{0\}$, such that the properties specified in Section II and the introduction of Section III are all met.

Definition 5. We use $\mathcal{H}$ ($\mathcal{H} \subset \mathcal{V}$) to define $n+1$ agents in $\mathbb{R}^n$ positioned at vertices of an n-D simplex in $\mathbb{R}^n$.

Definition 6. Set $\mathcal{CONT}(\mathcal{H})$ ($\mathcal{CONT}(\mathcal{H}) \subset \mathcal{V}$) defines all agents that are inside the convex hull specified by $\mathcal{H}$. To be more precise, $\mathcal{H}$ defines $n+1$ agents at vertices of an n-D simplex enclosing $\mathcal{CONT}(\mathcal{H})$.

Definition 7. Given $\mathcal{H}$,

$$c(\mathcal{H}) = \underset{j \in \mathcal{CONT}(\mathcal{H})}{\operatorname{argmin}} \left(\sum_{h \in \mathcal{H}} \|\mathbf{a}_h - \mathbf{a}_j\| \right) \tag{22}$$

is the agent belonging to $\mathcal{CONT}(\mathcal{H})$, at the closest distance from every member of $\mathcal{H}$, and called *mentee of* $\mathcal{H}$.

Definition 8. Given $\mathcal{H}$, as vertices of an n-D simplex in $\mathbb{R}^n$, and $c(\mathcal{H})$ as the mentee of $\mathcal{H}$,

$$\text{EXPAND}(\mathcal{H}, c(\mathcal{H})) = \mathcal{H}_1(\mathcal{H}, c(\mathcal{H})) \cup \cdots \cup \mathcal{H}_{n+1}(\mathcal{H}, c(\mathcal{H})) \tag{23}$$

decomposes the convex hull defined by $\mathcal{H}$ into $n+1$ cells all sharing vertex $c(\mathcal{H})$, where $\mathcal{H}_j$ defines $n+1$ vertices of the j-th simplex cell of $\mathcal{H}$, for $j = 1, \cdots, n+1$.

For better clarification, Fig. 3(a) shows reference configuration of an agent team in a 2-D motion space ($x-y$ plane) where $\mathcal{H} = \{10, 15, 18\}$ defines agents at vertices of a 2-D simplex (triangle). Set $\mathcal{CONT}(\mathcal{H}) = \{1, 5, 12, 16\}$ defines the agents contained by this triangle. As seen, $c(\mathcal{H}) = 5$ is the mentee of $\mathcal{H}$. Also, $\text{EXPAND}(\{10, 15, 18\}, 5) = \mathcal{H}_1 \cup \mathcal{H}_2 \cup \mathcal{H}_3$, where $\mathcal{H}_1 = \{10, 18, 5\}$, $\mathcal{H}_2 = \{18, 15, 5\}$, and $\mathcal{H}_3 = \{15, 10, 5\}$.

We apply Algorithm 1 to obtain set $\mathcal{E}$, as graphically shown in 3(b), given an arbitrary initial distribution of the agent team. We then convert the communication graph shown in Fig. 3(a) into the coverage neural network of Fig. 3(c) with 4 layers identified by set $\mathcal{M} = \{0, 1, 2, 3\}$, where $\mathcal{W}_0 = \mathcal{V}_0$ (defining the boundary and core leaders) has no in-neighbors, $\mathcal{W}_1 = \mathcal{W}_0 \bigcup \mathcal{V}_1$, $\mathcal{W}_2 = \mathcal{W}_1 \bigcup \mathcal{V}_2$, and $\mathcal{W}_1 = \mathcal{V}_3$. Note that $\mathcal{W}_0$ identifies the neurons that represent boundary agents.

B. Abstract Representation of Target Locations

We use the approach presented in Algorithm 2 to abstractly represent target set $\mathcal{D}$ by position vectors $\mathbf{p}_1, \cdots, \mathbf{p}_N$, given (i) desired positions of boundary and core agents, denoted by $\mathbf{p}_{b_1}$ through $\mathbf{p}_{b_{N_B+1}}$, (ii) the edge set $\mathcal{E}$, and (iii) target set $\mathcal{D}$, as the input. Note that $\mathbf{p}_i$ is considered the global desired position of $i \in \mathcal{V} \setminus \mathcal{V}_0$, but no follower $i \in \mathcal{V} \setminus \mathcal{V}_0$ knows $\mathbf{p}_i$. The desired position of $i \in \mathcal{V} \setminus \mathcal{V}_0$ is obtained by

$$\mathbf{p}_i = \frac{\sum_{h \in \bar{\mathcal{D}}_i} \mathbf{d}_h}{|\bar{\mathcal{D}}_i|}, \qquad \forall i \in \mathcal{V} \setminus \mathcal{V}_0, \tag{24}$$

where $\bar{\mathcal{D}}_i$, defined by Eq. (20), is a target data subset that is enclosed by $\bar{C}_i$ and defined by Eq. (18a).

Given desired positions of every follower agent $i \in \mathcal{V} \setminus \mathcal{V}_0$ and every in-neighbor agent $j \in \mathcal{N}(i)$, $\varpi_{i,j} > 0$ defines **the desired communication weight** between $i \in \mathcal{V} \setminus \mathcal{W}_0$ and $j \in \mathcal{N}(i)$, and is obtained by solving $n+1$ linear algebraic equations provided by

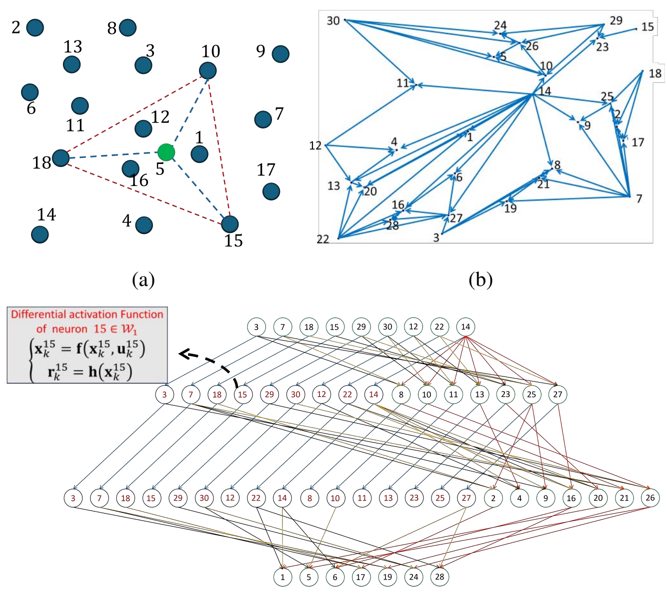

Differential activation Function of neuron $15 \in \mathcal{W}_1$
$\begin{cases}\mathbf{x}_k^{15} = \mathbf{f}(\mathbf{x}_k^{15}, \mathbf{u}_k^{15}) \\ \mathbf{r}_k^{15} = \mathbf{h}(\mathbf{x}_k^{15})\end{cases}$
(a)
(b)
(c)

Fig. 3: (a) Graphical representation of a 2-D simplex (triangle) with vertices defined by $\mathcal{H} = \{11, 15, 18\}$, where $\mathcal{CONT}(\mathcal{H}) = \{1, 5, 12, 16\}$ specifies all agents contained by this triangle and $c(\mathcal{H}) = 5$ is the mentee of $\mathcal{H}$. (b) An agent team with an arbitrary initial formation in a 2-D motion space and the corresponding communication graph obtained by Algorithm 1. (c) The DNN corresponding to the communication graph shown in Fig. 3 (b).

Algorithm 2 Assignment of followers' "desired" communication weights and "desired" positions.

```
1:  Get: Dataset $\mathcal{D}$; set $\mathcal{W}_0, \cdots, \mathcal{W}_M$; and graph $\mathcal{G}(\mathcal{V}, \mathcal{E})$.
2:  Obtain: Desired position of every agent $i \in \mathcal{V} \setminus \mathcal{W}_0$ and
    desired communication weights.
3:  for $l \in \mathcal{M} \setminus \{0\}$ do
4:      for $i \in$ do
5:          Obtain $\bar{\mathcal{D}}_i$ by using Eq. (20).
6:          Choose $\mathbf{p}_i$ as the geometric centroid of $\bar{\mathcal{D}}_i$.
7:          Get desired positions of in-neighbor agents $\mathcal{I}(i, l)$.
8:          Compute desired communication weights by (25).
9:      end for
10: end for
```

$$\mathbf{p}_i = \sum_{j \in \mathcal{I}(i,l)} \varpi_{i,j} \mathbf{p}_j, \qquad i \in \mathcal{V}_l,\ l \in \mathcal{M}, \tag{25a}$$

$$\sum_{j \in \mathcal{I}(i,l)} \varpi_{i,j} = 1, \qquad i \in \mathcal{V}_l,\ l \in \mathcal{M}. \tag{25b}$$

Definition 9. We define desired communication weight matrix $\mathbf{\Gamma}$ where $\mathbf{\Gamma} = [\Gamma_{ij}(t)] \in \mathbb{R}^{N \times N}$ is non-negative, and its (i, j) entry is given by

$$\Gamma_{ij} = \begin{cases} \varpi_{i,j} & i \in \mathcal{V} \setminus \mathcal{V}_0,\ j \in \mathcal{N}(i) \\ 0 & \text{otherwise} \end{cases}. \tag{26}$$

Proposition 1. *Constant desired weight $\varpi_{i,j}$ is unique for every agent $i \in \mathcal{V} \setminus \mathcal{W}_0$ with in-neighbor $j \in \mathcal{N}(i)$ if the target set $\mathcal{D}$ is time-invariant.*

Proof. If $\mathcal{D}$ is time-invariant, then $\mathbf{p}_i$ is constant for every $i \in \mathcal{V}$. Because in-neighbors of every agent $i \in \mathcal{V} \setminus \mathcal{W}_0$ are positioned at vertices of n-D simplex, Eqs. (25a) and (25b) provides $n+1$ linear algebraic equations with $n+1$ unknowns $\varpi_{i,j}$ that are uniquely determined. □

C. Learning the DNN Weights

For a decentralized coverage, it is necessary that every follower agent $i \in \mathcal{V}_l$, represented by a neuron in layer $l \in \mathcal{M} \setminus \{0\}$, chooses control $\mathbf{u}_i \in \mathbb{R}^{n_u \times 1}$, based on actual positions of the in-neighbor agents $\mathcal{N}(i)$, such that $\mathbf{r}_i(t)$ stably tracks $\mathbf{r}_{i,d}(t)$ defined by Eq. (9). To assign $\mathbf{r}_{i,d}(t)$, communication weight $w_{i,j}(t)$ between agent $i \in \mathcal{V}_l$ and every in-neighbor agent $j \in \mathcal{N}(i)$ needs to be learned at any time t.

We use a "forward" method to obtain the followers' communication weights (train the DNN weights) at any time t. This means that communication weights of layer $l \in \mathcal{M} \setminus \{0\}$ neurons are assigned before communication weights of layer $l+1 \in \mathcal{M} \setminus \{0, M\}$ neurons. To implement this "forward" method, communication weights of every agent $i \in \mathcal{V}_l$ are assigned such that the desired position $\mathbf{r}_{i,d}$ is at the geometric centroid

$$\mathbf{c}_i(t) = \frac{\sum_{h \in \mathcal{D}_i(t)} \mathbf{d}_h(t)}{|\mathcal{D}_i(t)|}, \qquad i \in \mathcal{V}_l,\ l \in \mathcal{M} \setminus \{0\}, \tag{27}$$

of all target data defined by $\mathcal{D}_i$ at any time t. Note that target data points defined by $\mathcal{D}_i$ are all contained by n-D simplex $C_i(t)$, where vertices of $C_i(t)$ are occupied by actual positions of $\mathcal{N}(i)$'s agents. Therefore, $\mathbf{c}_i(t)$ is inside $C_i(t)$ at any time t and communication of agent $i \in \mathcal{V}_l$ with in-neighbor $j \in \mathcal{N}(i)$, denoted by $w_{i,j}(t)$, is unique and obtained by solving the following $n+1$ algebraic equations [30]:

$$\mathbf{c}_i(t) = \sum_{j \in \mathcal{I}(i,l)} w_{i,j}(t)\mathbf{r}_j(t), \qquad i \in \mathcal{V}_l,\ l \in \mathcal{M}, \tag{28a}$$

$$\sum_{j \in \mathcal{I}(i,l)} w_{i,j}(t) = 1, \qquad i \in \mathcal{V}_l,\ l \in \mathcal{M}. \tag{28b}$$

To be more precise, communication weights obtained by Eq. (28) are unique because $\mathcal{N}(i)$'s agents are positioned at vertices of n-D simplex in $\mathbb{R}^n$. Also, agent i's communication weights are all positive because the geometric centroid $\mathbf{c}_i(t)$ is inside $C_i(t)$ at any time t (the proofs for uniqueness and positiveness of followers' communication weights under n-D polyhedralization are provided in Refs. [30], [31]).

Definition 10. We define weighted communication matrix

$$\mathbf{\Lambda}(t) = -\mathbf{I} + \mathbf{W}(t), \tag{29}$$

where $\mathbf{W}(t) = \left[W_{ij}(t)\right] \in \mathbb{R}^{N \times N}$ is non-negative, and its (i,j) entry is given by

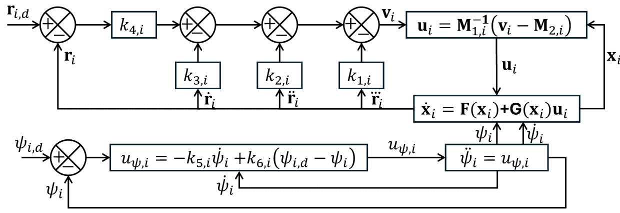
$$W_{ij}(t) = \begin{cases} w_{i,j}(t) & i \in \mathcal{V} \setminus \mathcal{V}_0,\ j \in \mathcal{N}(i) \\ 0 & \text{otherwise} \end{cases}. \tag{30}$$

Theorem 1. *Assume every agent $i \in \mathcal{V}$ chooses control input $\mathbf{u}_i$ such that $\mathbf{r}_i(t)$ asymptotically tracks $\mathbf{r}_{i,d}(t)$, defined by Eq. (9). Then, $\mathbf{r}_i(t)$ asymptotically converges to the desired position $\mathbf{p}_i$ for every $i \in \mathcal{V}$.*

Proof. If every agent $j \in \mathcal{V}_0$ asymptotically tracks $\mathbf{r}_{j,d}(t)$, then, actual position $\mathbf{r}_j$ converges to $\mathbf{p}_j$ because $\mathbf{r}_{j,d}(t) = \mathbf{p}_j$ is constant per Eq. (9). Then, for every $i \in \mathcal{W}_1 \setminus \mathcal{W}_0$, vertices of the simplex $\bar{C}_i$, belonging to $\mathcal{W}_0$, asymptotically converge to the vertices $\bar{C}_i$, where $\bar{C}_i$ and C_i enclose target data subsets $\bar{\mathcal{D}}_i$ and $\mathcal{D}_i$, respectively. This implies that $\mathbf{r}_{i,d}(t)$, defined as the geometric centroid of $\mathcal{D}_i(t)$ asymptotically converges to $\mathbf{p}_i$ for every $i \in \mathcal{W}_1$. By extending this logic, we can say that this convergence is propagated through the feedforward network. As the result, for every agent $i \in \mathcal{W}_l$ and layer $l \in \mathcal{M} \setminus \{0\}$, vertices of simplex $C_i(t)$ asymptotically converge to the vertices of $\bar{C}_i$, which in turn implies that $\mathbf{r}_{i,d}(t)$ asymptotically converges to $\mathbf{p}_i$. This also implies that $\mathbf{r}_i$ asymptotically converges to $\mathbf{p}_i$ per the theorem's assumption. □

IV. Multi-Agent Coverage Dynamics, Stability Analysis, and Convergence

In this section, we will assume that each agent is a quadcopter and apply the input-state feedback linearization described in [32], [33] for trajectory tracking using the block diagram depicted in Fig. 4. Therefore, the quadcopter dynamics presented in [32], [33] are used to operate the neurons of the DNN.



$\mathbf{r}_{i,d}$
$\mathbf{r}_i$
$k_{4,i}$
$k_{3,i}$
$k_{2,i}$
$k_{1,i}$
$\dot{\mathbf{r}}_i$
$\ddot{\mathbf{r}}_i$
$\dddot{\mathbf{r}}_i$
$\mathbf{v}_i$
$\mathbf{u}_i = \mathbf{M}_{1,i}^{-1}(\mathbf{v}_i - \mathbf{M}_{2,i})$
$\mathbf{u}_i$
$\mathbf{x}_i$
$\dot{\mathbf{x}}_i = \mathbf{F}(\mathbf{x}_i) + \mathbf{G}(\mathbf{x}_i)\mathbf{u}_i$
$\psi_{i,d}$
$u_{\psi,i} = -k_{5,i}\dot{\psi}_i + k_{6,i}(\psi_{i,d} - \psi_i)$
$u_{\psi,i}$
$\ddot{\psi}_i = u_{\psi,i}$
ψ_i
$\dot{\psi}_i$

Fig. 4: The block digram of the control system of every quadcopter $i \in \mathcal{V}$.

A. Quadcopter Dynamics and Trajectory Tracking

To model quadcopter motion, we use x_i, y_i, and z_i to define quadcopter i's position components. We also use ϕ_i, θ_i, and ψ_i to define the roll, pitch, and yaw angles, where the 3-2-1 Euler angle standard is applied to characterize the orientation of quadcopter $i \in \mathcal{V}$. Additionally, the thrust force magnitude produced by the quadcopter $i \in \mathcal{V}$ is denoted by p_i at time t. The dynamics of quadcoper $i \in \mathcal{V}$ is presented in the form of Eq. (8), where $\mathbf{f}_i(\mathbf{x}_i, \mathbf{u}_i) = \mathbf{F}(\mathbf{x}_i) + \mathbf{G}(\mathbf{x}_i)\mathbf{u}_i$, for every $i \in \mathcal{V}$,

$$\mathbf{x}_i = \begin{bmatrix} x_i & y_i & z_i & \dot{x}_i & \dot{y}_i & \dot{z}_i & \phi_i & \theta_i & \psi_i & \dot{\phi}_i & \dot{\theta}_i & \dot{\psi}_i & p_i & \dot{p}_i \end{bmatrix}^T$$

$$\mathbf{u}_i = \begin{bmatrix} u_{1,i} & u_{2,i} & u_{3,i} & u_{4,i} \end{bmatrix}^T, \tag{31}$$

$$\mathbf{F}(\mathbf{x}_i)=\begin{bmatrix}\dot{x}_i\\ \dot{y}_i\\ \dot{z}_i\\ \frac{p_i}{m}\left(\sin\phi_i\sin\psi_i+\cos\phi_i\cos\psi_i\sin\theta_i\right)\\ \frac{p_i}{m}\left(\cos\phi_i\sin\psi_i\sin\theta_i-\sin\phi_i\cos\psi_i\right)\\ \frac{p_i}{m}\cos\phi_i\cos\theta_i-9.81\\ \dot{\phi}_i\\ \dot{\theta}_i\\ \dot{\psi}_i\\ 0\\ 0\\ 0\\ \dot{p}_i\\ 0\end{bmatrix}, \quad (32)$$

$$\mathbf{G}(\mathbf{x}_i)=\begin{bmatrix}\mathbf{g}_1 & \mathbf{g}_2 & \mathbf{g}_3 & \mathbf{g}_4\end{bmatrix}=\begin{bmatrix}\mathbf{0}_{9\times1} & \mathbf{0}_{9\times3}\\ \mathbf{0}_{3\times1} & \mathbf{I}_3\\ 0 & \mathbf{0}_{1\times3}\\ 1 & \mathbf{0}_{1\times3}\end{bmatrix}, \quad (33)$$

By defining transformation $\mathbf{x}_i \rightarrow \left(\mathbf{r}_i,\dot{\mathbf{r}}_i,\ddot{\mathbf{r}}_i,\dddot{\mathbf{r}}_i,\psi_i,\dot{\psi}_i\right)$, we can use the input-state feedback linearization approach presented in [32] and convert the quadcopter dynamics to the following external dynamics [32], [34]:

$$\ddddot{\mathbf{r}}_i=\mathbf{v}_i, \quad (34a)$$

$$\ddot{\psi}_i=u_{\psi,i}, \quad (34b)$$

where $\mathbf{v}_i$ is related to the control input of quadcopter $i\in\mathcal{V}$, denoted by $\mathbf{u}_i$, by [35]

$$\mathbf{v}_i=\mathbf{M}_{1,i}\mathbf{u}_i+\mathbf{M}_{2,i}, \quad (35)$$

with

$$\mathbf{M}_{1,i}=\begin{bmatrix}L_{\mathbf{g}_1}L_{\mathbf{f}}^3x_i & L_{\mathbf{g}_2}L_{\mathbf{f}}^3x_i & L_{\mathbf{g}_3}L_{\mathbf{f}}^3x_i & L_{\mathbf{g}_4}L_{\mathbf{f}}^3x_i\\ L_{\mathbf{g}_1}L_{\mathbf{f}}^3y_i & L_{\mathbf{g}_2}L_{\mathbf{f}}^3y_i & L_{\mathbf{g}_3}L_{\mathbf{f}}^3y_i & L_{\mathbf{g}_4}L_{\mathbf{f}}^3y_i\\ L_{\mathbf{g}_1}L_{\mathbf{f}}^3z_i & L_{\mathbf{g}_2}L_{\mathbf{f}}^3z_i & L_{\mathbf{g}_3}L_{\mathbf{f}}^3z_i & L_{\mathbf{g}_4}L_{\mathbf{f}}^3z_i\\ L_{\mathbf{g}_1}L_{\mathbf{f}}\psi_i & L_{\mathbf{g}_2}L_{\mathbf{f}}\psi_i & L_{\mathbf{g}_3}L_{\mathbf{f}}\psi_i & L_{\mathbf{g}_4}L_{\mathbf{f}}\psi_i\end{bmatrix}\in\mathbb{R}^{14\times14}, \quad (36a)$$

$$\mathbf{M}_{2,i}=\begin{bmatrix}L_{\mathbf{f}}^4x_i & L_{\mathbf{f}}^4y_i & L_{\mathbf{f}}^4z_i & L_{\mathbf{f}}^2\psi_i\end{bmatrix}^T\in\mathbb{R}^{14\times1}. \quad (36b)$$

where the entries of $\mathbf{M}_{1,i}$ and $\mathbf{M}_{2,i}$ are the Lie derivative terms.

In this paper, we assume that the desired yaw angle is zero at any time t, and therefore, we choose

$$u_{\psi,i}=-k_{5,i}\dot{\psi}_i-k_{6,i}\psi_i, \quad (37)$$

where $k_{5,i}$ and $k_{6,i}$ are both positive. Hence, we can assume that $\psi_i(t)=0$ at any time t, and as a result, the quadcopter $i\in\mathcal{V}$ can be modeled by Eq. (34a). Here, we propose to choose $\mathbf{v}_i$ as follows:

$$\mathbf{v}_i=-k_{1,i}\dddot{\mathbf{r}}_i-k_{2,i}\ddot{\mathbf{r}}_i-k_{3,i}\dot{\mathbf{r}}_i+k_{4,i}\left(\mathbf{r}_{i,d}-\mathbf{r}_i\right), \qquad i\in\mathcal{V}, \quad (38)$$

where $\mathbf{r}_{i,d}(t)$ is defined by Eq. (9).

B. Multi-Agent Coverage Stability and Convergence

The quadcopter dynamics can be represented by the external dynamics of the quadcopter team, given by Eq. (34a) since the quadcopter dynamics is input-state feedback linerizable [32], [34]. Therefore, the network dynamics of the quadcopter team is obtained by [32], [35]

$$\frac{d}{dt}\left(\begin{bmatrix}\mathbf{Y}\\ \dot{\mathbf{Y}}\\ \ddot{\mathbf{Y}}\\ \dddot{\mathbf{Y}}\end{bmatrix}\right)=\mathbf{A}_{\text{MQS}}\begin{bmatrix}\mathbf{Y}\\ \dot{\mathbf{Y}}\\ \ddot{\mathbf{Y}}\\ \dddot{\mathbf{Y}}\end{bmatrix}+\mathbf{B}_{\text{MQS}}\mathbf{Z}, \quad (39)$$

where

$$\mathbf{Y}=\text{vec}\left(\begin{bmatrix}\mathbf{r}_1 & \cdots & \mathbf{r}_N\end{bmatrix}^T\right),$$

$$\mathbf{Z}=\text{vec}\left(\begin{bmatrix}\mathbf{p}_1 & \cdots & \mathbf{p}_N\end{bmatrix}^T\right),$$

$$\mathbf{A}_{\text{MQS}}=\begin{bmatrix}\mathbf{0}_{3N\times3N} & \mathbf{I}_{3N} & \mathbf{0}_{3N\times3N} & \mathbf{0}_{3N\times3N}\\ \mathbf{0}_{3N\times3N} & \mathbf{0}_{3N\times3N} & \mathbf{I}_{3N} & \mathbf{0}_{3N\times3N}\\ \mathbf{0}_{3N\times3N} & \mathbf{0}_{3N\times3N} & \mathbf{0}_{3N\times3N} & \mathbf{I}_{3N}\\ \mathbf{I}_3\otimes(\mathbf{K}_4\mathbf{\Lambda}(t)) & -\mathbf{K}_3\mathbf{I}_{3N} & -\mathbf{K}_2\mathbf{I}_{3N} & -\mathbf{K}_1\mathbf{I}_{3N}\end{bmatrix},$$

$$\mathbf{B}_{\text{MQS}}=\begin{bmatrix}\mathbf{0}_{9N\times3N}\\ \mathbf{I}_3\otimes(\mathbf{K}_4\mathbf{L})\end{bmatrix},$$

$$\mathbf{K}_j=\text{diag}\left(k_{j,1},\cdots,k_{j,N}\right), \qquad j=1,2,3,4,$$

$\mathbf{I}_{3N}\in\mathbb{R}^{3N\times3N}$ is the identity matrix, $\mathbf{0}_{3N\times3N}\in\mathbb{R}^{3N\times3N}$ is a zero-entry matrix, "$\otimes$" is Kronecker product symbol, and "vec" is the matrix vectorization operator. Also, $\mathbf{L}=\left[l_{ij}\right]\in\mathbb{R}^{N\times N}$ is defined by

$$l_{ij}=\begin{cases}1 & i\in\mathcal{V}_1,\ j\in\mathcal{N}(i)\\ 0 & \text{otherwise}\end{cases}. \quad (40)$$

The quadcopter team collective dynamics, given by Eq. (39), is linear and time-varying because $\mathbf{\Lambda}(t)=-\mathbf{I}+\mathbf{W}(t)$ and $\mathbf{W}(t)\in\mathbb{R}^{N\times N}$ is a time-varying weight matrix. Therefore, it is difficult to choose control gain matrices $\mathbf{K}_1$ through $\mathbf{K}_4$ to ensure stability of quadcopter team coordination if we consider dynamics Eq. (39) for analysis. In order to tackle this problem, we first use Theorem 2 below to convert dynamics (39) to a set of chained linear time-invariant dynamics with bounded inputs. Then, we provide conditions on the stability and convergence of the agent team dynamics in Theorem 4.

Definition 11. Assuming $|\mathcal{V}_l|=N_l$, for every $l\in\mathcal{M}$, O_l assigns a unique order number to every $\mathcal{V}_l$'s agent, where $N_{-1}=0$. The order number of agent $i\in\mathcal{V}_l$ is denoted by o_i, where $o_i=O_l(i)$, for every $l\in\mathcal{M}$.

Definition 12. We define transformation matrix $\mathbf{Q}_l=\left[Q_{ij}^l\right]\in\mathbb{R}^{N_l\times N}$ with (i,j) entry

$$Q_{ij}^l=\begin{cases}1 & j=O_l(i),\ i\in\mathcal{V}_l\\ 0 & \text{else}\end{cases}, \qquad l\in\mathcal{M}. \quad (41)$$

We note that

$$\mathbf{Q}_l\mathbf{Q}_l^T=\mathbf{I}_{N_l}, \qquad \forall l\in\mathcal{M}, \quad (42)$$

where $\mathbf{I}_{N_l}\in\mathbb{R}^{N_l\times N_l}$ is the identity matrix.

Definition 13. We define

$$\bar{\mathbf{Z}}_l=\left(\mathbf{I}_3\otimes\mathbf{Q}_l\right)\mathbf{Z}\in\mathbb{R}^{3N_l\times1}, \qquad l\in\mathcal{M}, \quad (43)$$

as the vector aggregating desired position components of all agents defined by $\mathcal{V}_l$ when agents' order numbers are defined by O_l, for every $l\in\mathcal{M}$.

Definition 14. We define

$$\bar{\mathbf{Y}}_l=\left(\mathbf{I}_3\otimes\mathbf{Q}_l\right)\mathbf{Y}\in\mathbb{R}^{3N_l\times1}, \qquad l\in\mathcal{M}, \quad (44)$$

as the vector aggregating actual position components of all agents defined by $\mathcal{V}_l$ when agents' order numbers are defined by O_l, for every $l\in\mathcal{M}$.

Definition 15. We define

$$\bar{\mathbf{S}}_l = \begin{bmatrix} \bar{\mathbf{Y}}_l^T & \dot{\bar{\mathbf{Y}}}_l^T & \ddot{\bar{\mathbf{Y}}}_l^T & \dddot{\bar{\mathbf{Y}}}_l^T \end{bmatrix}^T \in \mathbb{R}^{12N_l \times 1}, \qquad l \in \mathcal{M}, \tag{45}$$

to aggregate components of position, velocity, acceleration, and jerk of every agent belonging to $\mathcal{V}_l$ when agents' order numbers are defined by O_l, for every $l \in \mathcal{M}$.

Definition 16. We define

$$\bar{\mathbf{W}}_l(t) = \mathbf{Q}_l \mathbf{W}(t) \begin{bmatrix} \mathbf{Q}_{l-1}^T & \cdots & \mathbf{Q}_0^T \end{bmatrix}, \qquad l \in \mathcal{M} \setminus \{0\}, \tag{46}$$

as the **actual weight matrix aggregating actual communication weights** between $\mathcal{V}_l$'s agents and other agents belonging to $\mathcal{V}_0 \cup \cdots \cup \mathcal{V}_{l-1}$, for every $l \in \mathcal{M} \setminus \{0\}$.

$$\bar{W}_{o_i o_j}^l = \begin{cases} W_{ij}(t) & i \in \mathcal{V}_l,\ j \in \mathcal{N}(i) \subset \mathcal{V}_{l-1} \\ 0 & \text{otherwise} \end{cases}, \qquad l \in \mathcal{M} \setminus \{0\}. \tag{47}$$

Remark 2. Matrix $\bar{\mathbf{W}}_l \in \mathbb{R}^{N_l \times (N_0 + \cdots + N_{l-1})}$ is non-negative for $l \in \mathcal{M} \setminus \{0\}$, and sum of the entries of every row of $\bar{\mathbf{W}}_l$ is 1.

Definition 17. We define

$$\bar{\mathbf{\Gamma}}_l = \mathbf{Q}_l \mathbf{\Gamma} \begin{bmatrix} \mathbf{Q}_{l-1}^T & \cdots & \mathbf{Q}_0^T \end{bmatrix}, \qquad l \in \mathcal{M} \setminus \{0\}, \tag{48}$$

as the **desired weight matrix aggregating desired communication weights** between $\mathcal{V}_l$'s agents and other agents belonging to $\mathcal{V}_0 \cup \cdots \cup \mathcal{V}_{l-1}$, for every $l \in \mathcal{M} \setminus \{0\}$.

Definition 18. We define

$$\bar{\mathbf{K}}_{l,h} = \mathbf{Q}_l \mathbf{K}_h \mathbf{Q}_l^T \in \mathbb{R}^{N_l \times N_l}, \qquad l \in \mathcal{M},\ h = 1, \cdots, 4, \tag{49}$$

as a diagonal matrix aggregating the h-th control gains applied by $\mathcal{V}_l$'s agents, where $o_i = O_l(i)$ is the order number of $i \in \mathcal{V}_l$ and the (o_i, o_i) entry of $\bar{\mathbf{K}}_{l,h}$ is denoted by $K_{o_i o_i}^{l,h}$ and given by

$$K_{o_i o_i}^{l,h} = k_{i,h}, \qquad i \in \mathcal{V}_l,\ l \in \mathcal{M},\ h = 1, \cdots, 4. \tag{50}$$

Remark 3. Given $\mathbf{Q}_0$ and $\mathbf{L}$, defined by Eqs. (41) and (40), respectively, the following relation holds:

$$\mathbf{Q}_0 \mathbf{L} \mathbf{Q}_0^T = \mathbf{I}_{N_0}. \tag{51}$$

Definition 19. We define

$$\bar{\mathbf{R}}_l = \left(\mathbf{I}_3 \otimes \begin{bmatrix} \mathbf{Q}_{l-1} \\ \vdots \\ \mathbf{Q}_0 \end{bmatrix} \right) \mathbf{Y}. \tag{52}$$

Theorem 2. *The agent team collective dynamics can be converted to*

$$\begin{cases} \dot{\bar{\mathbf{S}}}_l = \mathbf{A}_l \bar{\mathbf{S}}_l + \mathbf{B}_l \bar{\mathbf{U}}_l \\ \bar{\mathbf{Y}}_l = \mathbf{C}_l \bar{\mathbf{S}}_l \end{cases}, \qquad \forall l \in \mathcal{M} \tag{53}$$

where $\bar{\mathbf{S}}_l \in \mathbb{R}^{12N_l \times 1}$ is state vector, $\bar{\mathbf{Y}}_l \in \mathbb{R}^{3N_l \times 1}$ is the output vector, and

$$\bar{\mathbf{U}}_l = \begin{cases} \bar{\mathbf{Z}}_l & l = 0 \in \mathcal{M} \\ \left(\mathbf{I}_3 \otimes \bar{\mathbf{W}}_l\right) \bar{\mathbf{R}}_l & l \in \mathcal{M} \setminus \{0\} \end{cases} \tag{54}$$

is the input vector. Also,

$$\mathbf{A}_l = \begin{bmatrix} \mathbf{0}_{3N_l \times 3N_l} & \mathbf{I}_{3N_l} & \mathbf{0}_{3N_l \times 3N_l} & \mathbf{0}_{3N_l \times 3N_l} \\ \mathbf{0}_{3N_l \times 3N_l} & \mathbf{0}_{3N_l \times 3N_l} & \mathbf{I}_{3N_l} & \mathbf{0}_{3N_l \times 3N_l} \\ \mathbf{0}_{3N_l \times 3N_l} & \mathbf{0}_{3N_l \times 3N_l} & \mathbf{0}_{3N_l \times 3N_l} & \mathbf{I}_{3N_l} \\ -\mathbf{I}_3 \otimes \bar{\mathbf{K}}_{l,4} & -\mathbf{I}_3 \otimes \bar{\mathbf{K}}_{l,3} & -\mathbf{I}_3 \otimes \bar{\mathbf{K}}_{l,2} & -\mathbf{I}_3 \otimes \bar{\mathbf{K}}_{l,1} \end{bmatrix}, \ l \in \mathcal{M}, \tag{55a}$$

$$\mathbf{B}_l = \begin{bmatrix} \mathbf{0}_{9N_l \times 3N_l} \\ \mathbf{I}_3 \otimes \bar{\mathbf{K}}_{l,4} \end{bmatrix}, \qquad l \in \mathcal{M}, \tag{55b}$$

$$\mathbf{C}_l = \begin{bmatrix} \mathbf{I}_{3N_l} & \mathbf{0}_{3N_l \times 9N_l} \end{bmatrix}, \qquad l \in \mathcal{M}, \tag{55c}$$

$\mathbf{I}_3 \in \mathbb{R}^{3\times3}$ *and* $\mathbf{I}_{3N_l} \in \mathbb{R}^{3N_l \times 3N_l}$ *are identity matrices;* $\mathbf{0}_{3N_l \times 9N_l} \in \mathbb{R}^{3N_l \times 9N_l}$, $\mathbf{0}_{9N_l \times 3N_l} \in \mathbb{R}^{9N_l \times 3N_l}$, $\mathbf{0}_{3N_l \times 3N_l} \in \mathbb{R}^{3N_l \times 3N_l}$*, and* $\mathbf{0}_{9N_l \times 9N_l} \in \mathbb{R}^{9N_l \times 9N_l}$ *are zero-entry matrices.*

Proof. The agent team dynamics (39) can be rewritten as follows:

$$\sum_{h=1}^{4} \left(\mathbf{I}_3 \otimes \mathbf{K}_h\right) \frac{d^{4-h}\mathbf{Y}}{dt^{4-h}} + \left(\mathbf{I}_3 \otimes \mathbf{K}_4 \mathbf{W}(t)\right) \mathbf{Y} + \left(\mathbf{I}_3 \otimes \mathbf{K}_4 \mathbf{L}\right) \mathbf{Z} = 0. \tag{56}$$

Because $\mathcal{V}_0$'s agents do not communicate with any other agent,

$$\mathbf{Q}_0 \mathbf{W}(t) = \mathbf{0}_{N_0 \times N} \in \mathbb{R}^{N_0 \times N}, \qquad 0 \in \mathcal{M}.$$

Also, entries of row h of matrix $\mathbf{L}$ are all zero, if $h \notin \mathcal{V}_0$. This implies that

$$\mathbf{Q}_l \mathbf{L}(t) = \mathbf{0}_{N_l \times N} \in \mathbb{R}^{N_l \times N}, \qquad \forall l \in \mathcal{M} \setminus \{0\}.$$

Therefore, pre-multiplying Eq. (56), by $\mathbf{I}_3 \otimes \mathbf{Q}_l$, Eq. (56) simplifies to

$$\sum_{h=1}^{4} \left(\mathbf{I}_3 \otimes \mathbf{Q}_0 \mathbf{K}_h\right) \frac{d^{4-h}\mathbf{Y}}{dt^{4-h}} + \left(\mathbf{I}_3 \otimes \mathbf{Q}_0 \mathbf{K}_4 \mathbf{L}\right) \mathbf{Z} = 0, \quad l = 0, \tag{57a}$$

$$\textstyle\sum_{h=1}^{4} \left(\mathbf{I}_3 \otimes \mathbf{Q}_l \mathbf{K}_h\right) \frac{d^{4-h}\mathbf{Y}}{dt^{4-h}} + \left(\mathbf{I}_3 \otimes \mathbf{Q}_l \mathbf{K}_4 \mathbf{W}(t)\right) \mathbf{Y} = 0, \qquad l \in \mathcal{M} \setminus \{0\}. \tag{57b}$$

When we use order number to sort agents, the following relations hold:

$$\begin{aligned} \left(\mathbf{I}_3 \otimes \mathbf{Q}_0 \mathbf{K}_4 \mathbf{L}\right) \mathbf{Z} &= \left(\mathbf{I}_3 \otimes \left(\mathbf{Q}_0 \mathbf{K}_4 \mathbf{Q}_0^T \mathbf{Q}_0 \mathbf{L} \mathbf{Q}_0^T\right)\right) \left(\mathbf{I}_3 \otimes \mathbf{Q}_0\right) \mathbf{Z} \\ &= \left(\mathbf{I}_3 \otimes \bar{\mathbf{K}}_{0,4} \mathbf{I}_{N_0}\right) \bar{\mathbf{Z}}_0 = \left(\mathbf{I}_3 \otimes \bar{\mathbf{K}}_{0,4}\right) \bar{\mathbf{Z}}_0, \end{aligned} \tag{AA}$$

$$\begin{aligned} \left(\mathbf{I}_3 \otimes \mathbf{Q}_l \mathbf{K}_h\right) \mathbf{Y} &= \left(\mathbf{I}_3 \otimes \left(\mathbf{Q}_l \mathbf{K}_h \mathbf{Q}_l^T\right) \left(\mathbf{I}_3 \otimes \mathbf{Q}_l\right)\right) \mathbf{Y} \\ &= \left(\mathbf{I}_3 \otimes \bar{\mathbf{K}}_{l,h}\right) \bar{\mathbf{Y}}_l, \end{aligned} \tag{BB}$$

$$\begin{aligned} &\left(\mathbf{I}_3 \otimes \mathbf{Q}_l \mathbf{K}_4 \mathbf{W}\right) \mathbf{Y} = \left(\mathbf{I}_3 \otimes \mathbf{Q}_l \mathbf{K}_4 \mathbf{Q}_l^T \mathbf{Q}_l \mathbf{W}\right) \left(\mathbf{I}_3 \otimes \begin{bmatrix} \mathbf{Q}_{l-1}^T & \cdots & \mathbf{Q}_0^T \end{bmatrix}\right) \left(\mathbf{I}_3 \otimes \begin{bmatrix} \mathbf{Q}_{l-1} \\ \vdots \\ \mathbf{Q}_0 \end{bmatrix} \right) \mathbf{Y} = \\ &\left(\mathbf{I}_3 \otimes \mathbf{Q}_l \mathbf{K}_4 \mathbf{W}\right) \mathbf{Y} = \left(\mathbf{I}_3 \otimes \mathbf{Q}_l \mathbf{K}_4 \mathbf{Q}_l^T \mathbf{Q}_l \mathbf{W} \begin{bmatrix} \mathbf{Q}_{l-1}^T & \cdots & \mathbf{Q}_0^T \end{bmatrix}\right) \left(\mathbf{I}_3 \otimes \begin{bmatrix} \mathbf{Q}_{l-1} \\ \vdots \\ \mathbf{Q}_0 \end{bmatrix} \right) \mathbf{Y} = \\ &\left(\mathbf{I}_3 \otimes \bar{\mathbf{K}}_{l,4} \bar{\mathbf{W}}_l\right) \bar{\mathbf{R}}_l. \end{aligned} \tag{CC}$$

Therefore, by substituting (AA), (BB), and (CC), Eqs. (57a) and (57b) simplify to

$$\sum_{h=1}^{4} \left(\mathbf{I}_3 \otimes \bar{\mathbf{K}}_{0,h}\right) \frac{d^{4-h}\bar{\mathbf{Y}}_0}{dt^{4-h}} + \left(\mathbf{I}_3 \otimes \bar{\mathbf{K}}_{0,4}\right) \bar{\mathbf{Z}}_l = 0, \qquad l = 0, \tag{58a}$$

$$\textstyle\sum_{h=1}^{4} \left(\mathbf{I}_3 \otimes \bar{\mathbf{K}}_{l,h}\right) \frac{d^{4-h}\bar{\mathbf{Y}}_l}{dt^{4-h}} + \left(\mathbf{I}_3 \otimes \bar{\mathbf{K}}_{l,4} \bar{\mathbf{W}}_l(t)\right) \bar{\mathbf{R}}_l = 0, \qquad l \in \mathcal{M} \setminus \{0\}. \tag{58b}$$

Now, we can rewrite Eqs. (58a) and (58b) in the state space form given by Eq. (53). □

Note that the equilibrium set of the agent team dynamics, given by Eq. (39), is achieved if the following condition holds:

$$\bigwedge_{l \in \mathcal{M}} \bigwedge_{i \in \mathcal{V}_l} \left(\mathbf{r}_i = \mathbf{r}_{i,d}\right). \tag{59}$$

When we use Eq. (53) to model the agent team dynamics, equilibrium is achieved if the following conditions hold:

$$\bar{\mathbf{Y}}_0 = \bar{\mathbf{Z}}_0, \tag{60a}$$

$$\bigwedge_{l\in\mathcal{M}\setminus\{0\}} \left(\bar{\mathbf{Y}}_l - \left(\mathbf{I}_3 \otimes \bar{\mathbf{W}}_l\right)\bar{\mathbf{R}}_l\right). \tag{60b}$$

Theorem 3. *At equilibrium condition, specified by Eqs.* (60a) *and* (60b), $w_{i,j} = \varpi_{i,j}$ *which in turn implies that* $\bar{\mathbf{W}}_l = \bar{\mathbf{\Gamma}}_l$, *for every* $l \in \mathcal{M}\setminus\{0\}$.

Proof. For $l = 0 \in \mathcal{M}$, $\mathbf{r}_{i,d} = \mathbf{p}_i$, therefore, we can conclude that $\mathbf{r}_{j,d} = \mathbf{p}_j$, for every $j \in \mathcal{V}_1$, if $\mathbf{r}_i = \mathbf{p}_i$, for every $i \in \mathcal{V}_0$. This is because $\mathbf{r}_{j,d}$ assigned by Eq. (27) is at the geometric centroid of distribution of the data subset $\mathcal{D}_j(t)$ and $\mathcal{D}_j = \bar{\mathcal{D}}_j$ when $\mathbf{r}_i = \mathbf{p}_i$, for every $i \in \mathcal{V}_0$, which in turn implies that $\mathbf{r}_{j,d}$ converges to $\mathbf{p}_j$ for every $j \in \mathcal{V}_1$. With the similar logic, we can claim that $\mathbf{r}_{j,d} = \mathbf{p}_j$ for every agent $j \in \mathcal{V}_l$, if $\mathbf{r}_i = \mathbf{p}_i$ for every agent $i \in \mathcal{V}_0 \cup \cdots \cup \mathcal{V}_{l-1}$, for $l \in \mathcal{M}\setminus\{0\}$.

If $\mathbf{r}_{i,d} = \mathbf{r}_i = \mathbf{p}_i$ for every $i \in \mathcal{V}$, communication weight $\varpi_{i,j}$, assigned by Eqs. (25a) and (25b), is identical to actual communication weight $w_{i,j}$, assigned by Eqs. (28a) and (28b). This implies that $\mathbf{W} = \mathbf{\Gamma}$ and $\bar{\mathbf{W}}_l = \bar{\mathbf{\Gamma}}_l$, for every $l \in \mathcal{M}\setminus\{0\}$. □

Theorem 4. *Assume every agent* $i \in \mathcal{V}$ *is modeled by dynamics* (34a), *and chooses control input* $\mathbf{v}_i$, *with control gain* $k_{1,i}$, $k_{2,i}$, $k_{3,i}$, *and* $k_{4,i}$, *such that matrices* $\mathbf{A}_0$, $\mathbf{A}_1$, $\cdots$, $\mathbf{A}_M$ *are all Hurwitz. Then,* $\mathbf{r}_i(t)$ *asymptotically converges to the desired position* $\mathbf{p}_i$ *for every* $i \in \mathcal{V}$.

Proof. Matrices $\mathbf{A}_l$ and $\mathbf{B}_l$ are time-invariant for every $l \in \mathcal{M}$. By choosing proper gain metrics $\bar{\mathbf{K}}_{l,h}$, for every $l \in \mathcal{M}$ and $h = 1, \cdots, 4$, we can ensure that $\mathbf{A}_l$ is Hurwitz. Additionally, $\bar{\mathbf{W}}_l$ is non-negative and one-sum row, for every $l \in \mathcal{M}\setminus\{0\}$ at any time t (see Remark 2). Therefore, we can guarantee that the coverage dynamics (53) is Bounded Input Bounded Output (BIBO) stable, which in turn implies that $\|\bar{\mathbf{S}}_l(t)\|$ is bounded at any time t, for every $l \in \mathcal{M}$. Now, assuming $\mathbf{A}_0$ is Hurwitz, $\bar{\mathbf{Y}}_0 \to \bar{\mathbf{Z}}_0$ as $t \to \infty$ which in turn implies $\|\mathbf{r}_i(t) - \mathbf{p}_i\| \to 0$ as $t \to \infty$, for every $i \in \mathcal{V}_0$. This also implies that $\bar{\mathbf{W}}_1 \to \bar{\mathbf{\Gamma}}_1$ and $\mathbf{r}_{j,d} \to \mathbf{p}_j$ for every $j \in \mathcal{V}_1$ per Theorem 3. By extending this logic, we can prove that $\bar{\mathbf{Y}}_l \to \bar{\mathbf{Z}}_l$ and $\bar{\mathbf{W}}_l \to \bar{\mathbf{\Gamma}}_l$ as $t \to \infty$, if $\mathbf{A}_0$, $\cdots \mathbf{A}_l$ are all Hurwitz, for every $l \in \mathcal{M}\setminus\{0\}$. This implies that $\|\mathbf{r}_i(t) - \mathbf{p}_i\| \to 0$ for every $i \in \mathcal{V}_l$ and $l \in \mathcal{M}$. □

We can apply pole placement to assign diagonal control gain matrices $\bar{\mathbf{K}}_{l,h}$ for every $h \in \{1, \cdots, 4\}$ and $l \in \mathcal{M}$ every matrix $\bar{\mathbf{A}}_l$ is Hurwitz.

V. Simulation Results

We consider an agent team consisting of 57 quadcopters with the reference configuration shown in Fig. 5, where we use the model and trajectory control presented in Refs. [32], [33] for multi-agent coverage simulation. Here quadcopters 1 through 4 defined by set $\mathcal{V}_B = \{1,2,3,4\}$ are the boundary leader agents; agent 5 defined by singleton $\mathcal{V}_C = \{5\}$ is the core leader; and the remaining agents defined by $\mathcal{V}_I = \{6, \cdots, 57\}$ are followers.

TABLE I: Agents' order numbers

$l \in \mathcal{M}$	0	0	0	0	0	1	1	1	1	2
$i \in \mathcal{V}_l$	1	2	3	4	5	6	19	32	45	7
$O_l(i)$	1	2	3	4	5	6	7	8	9	10
$l \in \mathcal{M}$	2	2	2	2	2	2	2	2	2	2
$i \in \mathcal{V}_l$	8	9	20	21	22	33	34	35	46	47
$O_l(i)$	11	12	13	14	15	16	17	18	19	20
$l \in \mathcal{M}$	2	3	3	3	3	3	3	3	3	3
$i \in \mathcal{V}_l$	48	10	11	12	13	14	15	16	17	18
$O_l(i)$	21	22	23	24	25	26	27	28	29	30
$l \in \mathcal{M}$	3	3	3	3	3	3	3	3	3	3
$i \in \mathcal{V}_l$	23	24	25	26	27	28	29	30	31	36
$O_l(i)$	31	32	33	34	35	36	37	38	39	40
$l \in \mathcal{M}$	3	3	3	3	3	3	3	3	3	3
$i \in \mathcal{V}_l$	37	38	39	40	41	42	43	44	49	50
$O_l(i)$	41	42	43	44	45	46	47	48	49	50
$l \in \mathcal{M}$	3	3	3	3	3	3	3	-	-	-
$i \in \mathcal{V}_l$	51	52	53	54	55	56	57	-	-	-
$O_l(i)$	51	52	53	54	55	56	57	-	-	-

A. DNN Structure

The inter-agent communications are directional and shown by blue vectors in Fig. 5. The communication graph is defined by $\mathcal{G}(\mathcal{V}, \mathcal{E})$ and converted into the neural network shown in Fig. 6 with four layers, thus, $\mathcal{M} = \{0,1,2,3\}$ ($M = 3$), and $\mathcal{V}$ can be expressed as $\mathcal{V} = \mathcal{V}_0 \bigcup \mathcal{V}_1 \bigcup \mathcal{V}_2 \bigcup \mathcal{V}_3$, where $\mathcal{V}_l$ is defined in Table I for every $l \in \mathcal{M}$. Note that $N_0 = |\mathcal{V}_0| = 5$, $N_1 = |\mathcal{V}_1| = 4$, $N_2 = |\mathcal{V}_2| = 12$, and $N_3 = |\mathcal{V}_3| = 36$. In Fig. 5, the agents represented by $\mathcal{W}_0$, $\mathcal{W}_1$, $\mathcal{W}_2$, and $\mathcal{W}_3$ are colored by cyan, red, green, and black, respectively.

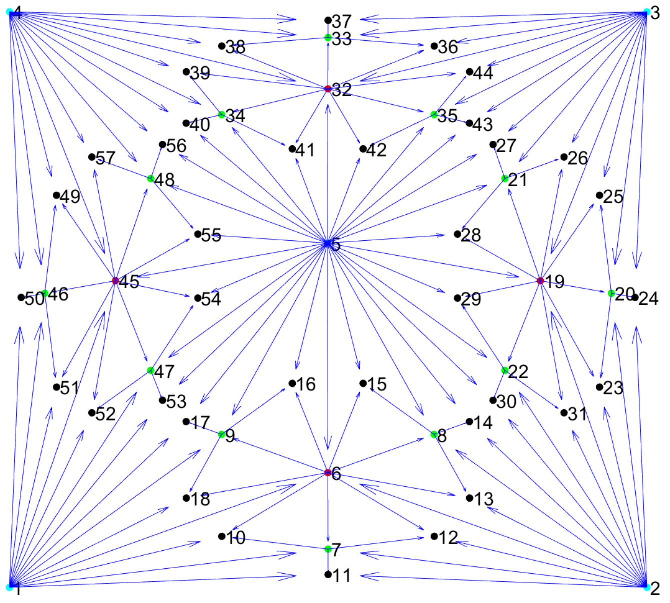


Fig. 5: Reference Configuration of the quadcopter team in a horizontal plane parallel to the $x-y$ plane. The inter-agent communication are directional and shown by blue arrays.

B. Abstract Representation of Target Data set

We apply the proposed coverage algorithm to cover elliptic, multi-circle, and triangular zones, each specified by the corresponding data set $\mathcal{D}$, where $\mathcal{D}$ defines 500 data points shown by green spots in Figs. 7 (a,b,c). As shown, each target set is

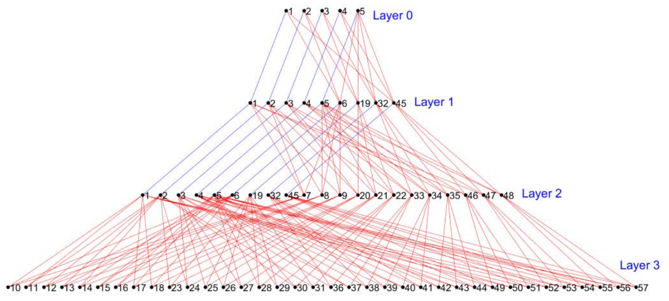


Fig. 6: The feed-forward network used for specifying inter-agent communication among the quadcopter team.

represented by 52 points positioned at $\mathbf{p}_6$ through $\mathbf{p}_{57}$, where they are obtained by using the approach presented in Section III-B. These points are shown by red in Figs. 7 (a,b,c).

C. Trajectory Tracking and Coverage Convergence

For simulation, we let every agent $i \in \mathcal{V}$ use the same control gain $k_1 = k_{1,i} = 7$, $k_2 = k_{2,i} = 17.75$, $k_3 = k_{3,i} = 19.25$, and $k_4 = k_{4,i} = 7.5$. Therefore, gain matrices $\mathbf{K}_1 = 7\mathbf{I}_{57}$, $\mathbf{K}_2 = 17.75\mathbf{I}_{57}$, $\mathbf{K}_3 = 19.25\mathbf{I}_{57}$, and $\mathbf{K}_4 = 7.5\mathbf{I}_{57}$ are obtained. By using Eq. (49),

$$\bar{\mathbf{K}}_{l,h} = k_h \mathbf{I}_{N_l}, \qquad l \in \mathcal{M},\ h = 1, \cdots, 4,$$

is obtained. Figure 8 shows the components of actual and desired positions of quadcopters 13, 45, and 51 plotted versus time overt time interval $[0, 20]s$, by solid black and dashed red, respectively. As seen, the actual positions of these three agents almost reached the desired positions at time $t = 12s$. Figure 9 shows the time-varying communication weights of agent 41 with its in-neighbors defined by $\mathcal{N}_{41} = \{34, 5, 32\}$. As shown, $w_{41,j}(t)$ converges to its desired value of $\varpi_{41,j}$ in about 12 seconds for every $j \in \mathcal{N}_{41}$. Decentralized coverage convergence is also demonstrated in Fig. 10 by showing the agent team configurations at different sample times when the target data is by a finite and discrete set over the triangular domain shown in Fig. 7 (c).

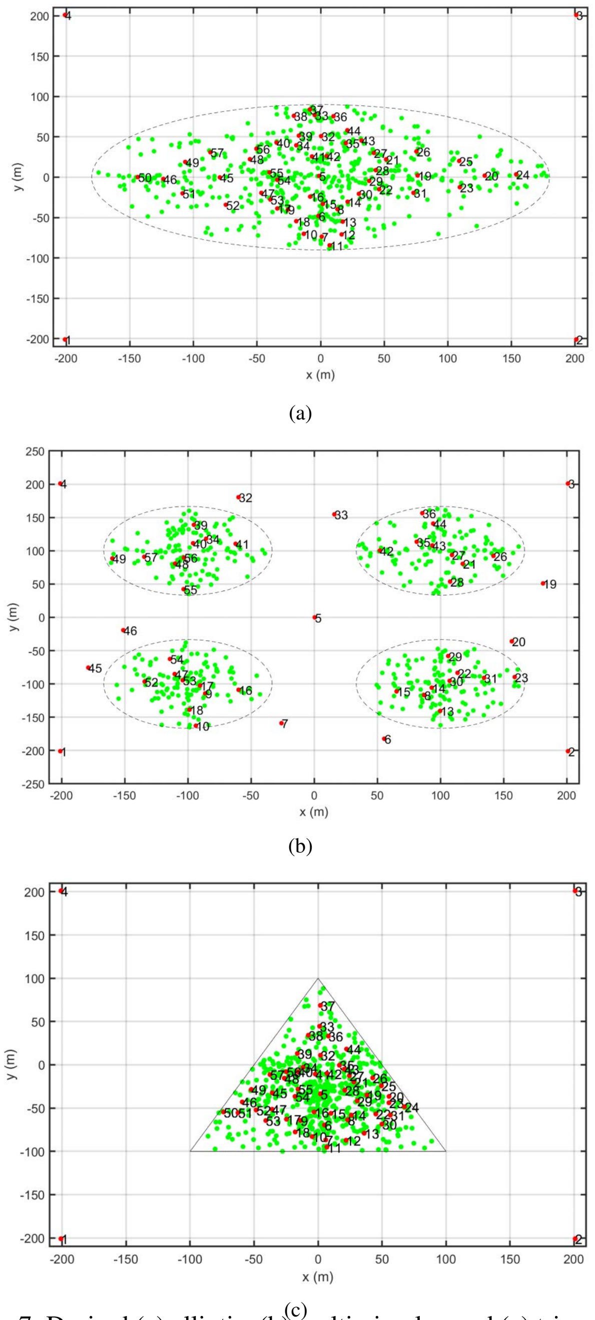


Fig. 7: Desired (a) elliptic, (b) multi-circular, and (c) triangular target distributions. The red spots are the abstract representation of the target sets by 52 nodes.

VI. Discussion

The paper assumes that the target data set is discrete, while available work on multi-agent coverage usually uses continuous mass density functions to define target distribution in the motion space. The proposed DNN-based can also achieve a decentralized distributed coverage when the target is defined by a smooth mass density function. For such a problem, we first obtain target set $\mathcal{D}'$ by uniform discretization of the motion space. Then, we use the Gaussian distribution to generate the intensity (mass density) function $\mathcal{T} : \mathcal{D}' \to (0, 1)$. More specifically, we define $\mathcal{D}' = \{1, \cdots, n'_d\}$ as n'_d nodes uniformly distributed in the motion space, where node $h \in \mathcal{D}'$ is positioned at $\mathbf{d}'_h$. The intensity function is then defined by

$$\mathcal{T}(\mathbf{r}) = \sum_{h \in \mathcal{D}'} \mathcal{N}\left(\mathbf{r}; \mathbf{d}'_h, \mathbf{\Sigma}'_h\right), \qquad \forall l \in \mathcal{D}', \tag{61}$$

where $\mathcal{N}\left(\mathbf{r}; \mathbf{d}'_h, \mathbf{\Sigma}'_h\right)$ is a multi-variate normal distribution specified by mean vector $\mathbf{d}'_h$ and covariance matrix $\mathbf{\Sigma}'_h$. Hence,

$$\bar{\mathbf{r}}_i = \frac{\int_{C_i} \mathcal{T}(\mathbf{r})\, dS}{\int_{C_i} \mathcal{T}(\mathbf{r})\, dS} \tag{62}$$

is the geometric centroid of subset set $\mathcal{D}_i(t) \subset \mathcal{D}$, and the DNN weights can be trained by solving sub-equations of Eq. (25).

VII. Conclusion and Future Work

We proposed a novel neural-network-based approach for multi-agent coverage of a target with unknown distribution. We

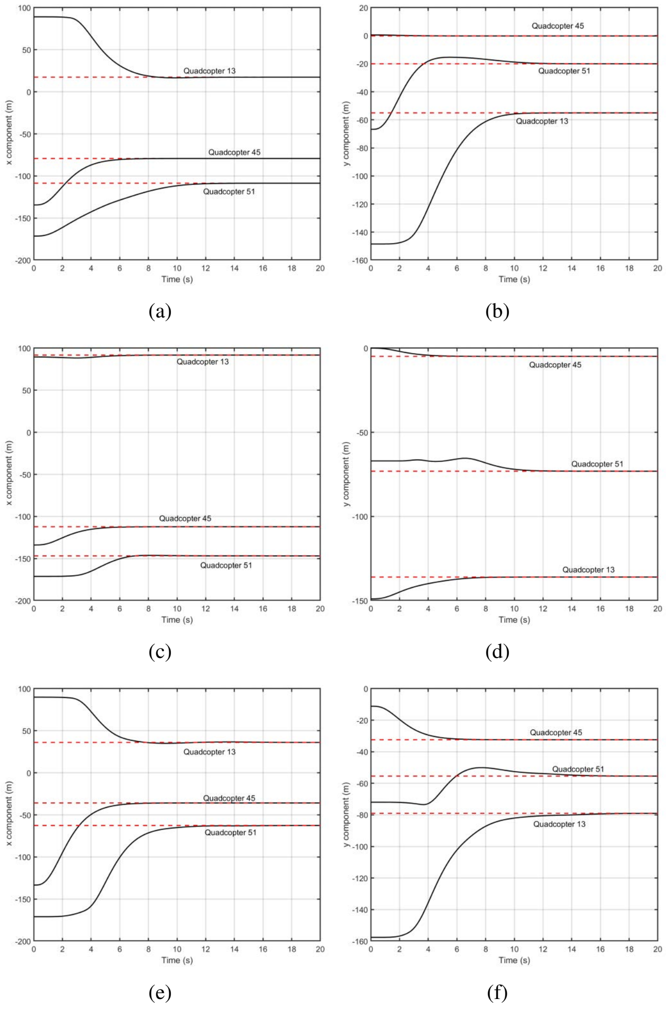


Fig. 8: (a,c,e) x components of actual position of agents 13, 45, and 51 over time interval $[0, 20]s$ shown by solid black curves. x components of desired position of agents 13, 45, and 51 over time interval $[0, 20]s$ shown by dashed red curves. (b,d,f) y components of actual position of agents 13, 45, and 51 over time interval $[0, 20]s$ shown by solid black curves. y components of desired position of agents 13, 45, and 51 over time interval $[0, 20]s$ shown by dashed red curves.

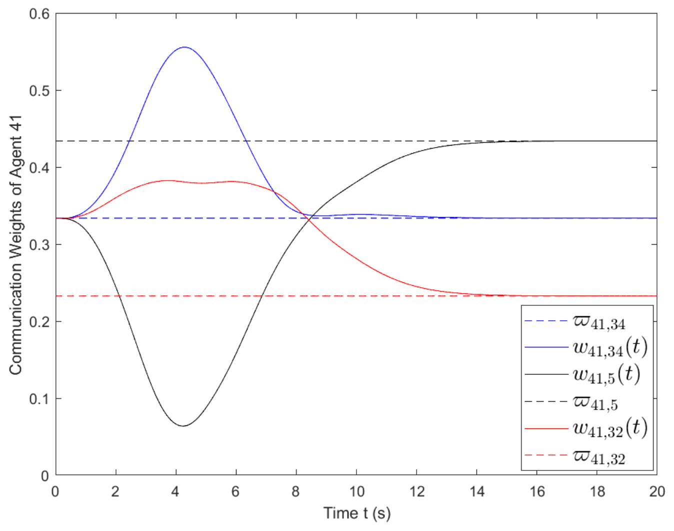


Fig. 9: Communication weights of agent 41 with in-neighbor agents 34, 5, and 32. The time varying communication weights $w_{41,34}(t)$, $w_{41,5}(t)$, and $w_{41,32}(t)$ converge to the desired values $\varpi_{41,34}$, $\varpi_{41,5}$, and $\varpi_{41,32}$ in about 12 seconds.

developed a forward approach to train the weights of the coverage neural network such that: (i) the target is represented by a finite number of points; (ii) the multi-agent system quickly and decentralizedly converges to the designated points representing the target distribution. The paper also developed a novel approach for structuring inter-agent communication based on agents' reference configurations arbitrarily distributed in $\mathbb{R}^n$. For validation, we performed a simulation of multi-agent coverage using a team of 57 quadcopters, each of which is represented by at least one neuron of the coverage neural network. The simulation results verified the fast and decentralized convergence of the proposed multi-agent coverage, where each quadcopter reached its designated position in about 12 seconds.

In this paper, we used the feedback linearization approach to design a nonlinear trajectory tacking control, where the eigen analysis method was applied to design the control gains. The Lyapunov direct method can also be used to assure stability and convergence of multi-agent coverage. Particularly, the outcome of Theorem 4 can be extended to DNN-based coverage by agents modeled by nonlinear dynamics if $\mathbf{u}_i$ is chosen such that the global stability of trajectory tracking is proven. For future work, we plan to extend the proposed DNN-based approach to search targets with a distribution given by a mass density function. Another future plan is to study DNN-based coverage of dynamic targets by providing guarantee conditions for the stability of multi-agent coordination and the decentralized convergence of the actual agent team configuration to a desired time-varying configuration that best represents a distributed target.

References

[1] R. N. Haksar, S. Trimpe, and M. Schwager, "Spatial scheduling of informative meetings for multi-agent persistent coverage," *IEEE Robotics and Automation Letters*, vol. 5, no. 2, pp. 3027–3034, 2020.
[2] E. Seraj and M. Gombolay, "Coordinated control of uavs for human-centered active sensing of wildfires," in *2020 American Control Conference (ACC)*, 2020, pp. 1645–1652.
[3] X. Pan, M. Liu, F. Zhong, Y. Yang, S.-C. Zhu, and Y. Wang, "Mate: Benchmarking multi-agent reinforcement learning in distributed target coverage control," *Advances in Neural Information Processing Systems*, vol. 35, pp. 27 862–27 879, 2022.
[4] A. Din, M. Y. Ismail, B. Shah, M. Babar, F. Ali, and S. U. Baig, "A deep reinforcement learning-based multi-agent area coverage control for smart agriculture," *Computers and Electrical Engineering*, vol. 101, p. 108089, 2022.
[5] M. Davoodi, S. Faryadi, and J. M. Velni, "A graph theoretic-based approach for deploying heterogeneous multi-agent systems with application in precision agriculture," *Journal of Intelligent & Robotic Systems*, vol. 101, pp. 1–15, 2021.
[6] E. Seraj, A. Silva, and M. Gombolay, "Multi-uav planning for cooperative wildfire coverage and tracking with quality-of-service guarantees," *Autonomous Agents and Multi-Agent Systems*, vol. 36, no. 2, p. 39, 2022.
[7] J. I. Vasquez and E. A. Merchán-Cruz, "A divide and conquer strategy for sweeping coverage path planning," *Energies*, vol. 15, no. 21, p. 7898, 2022.

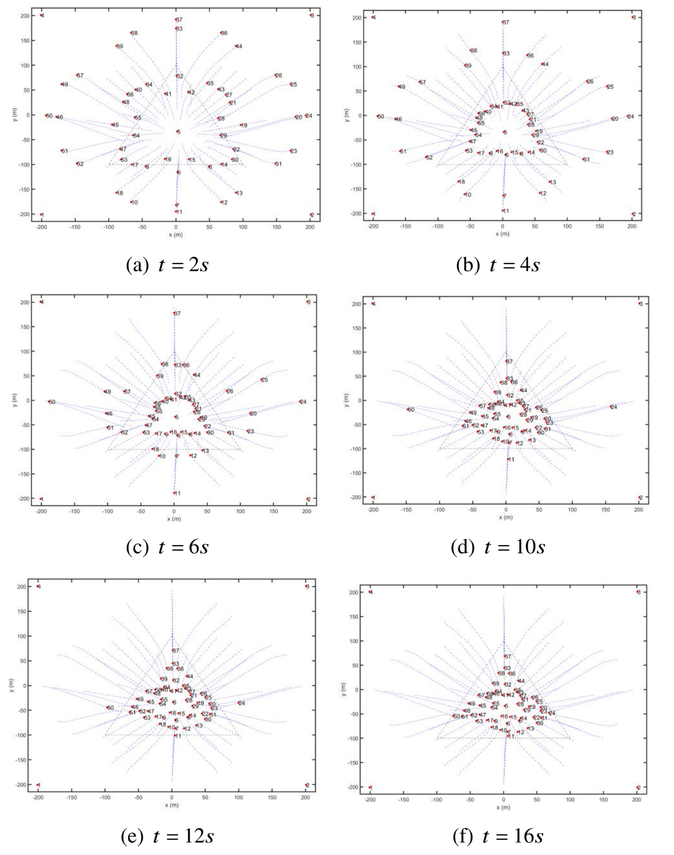

(a) $t = 2s$ (b) $t = 4s$ (c) $t = 6s$ (d) $t = 10s$ (e) $t = 12s$ (f) $t = 16s$

Fig. 10: Multi-agent configurations at different sample times when the desired target is distributed over the triangular domain shown in Fig. 7.

[8] S. Bochkarev and S. L. Smith, "On minimizing turns in robot coverage path planning," in *2016 IEEE international conference on automation science and engineering (CASE)*. IEEE, 2016, pp. 1237–1242.

[9] T. M. Cabreira, C. Di Franco, P. R. Ferreira, and G. C. Buttazzo, "Energy-aware spiral coverage path planning for uav photogrammetric applications," *IEEE Robotics and automation letters*, vol. 3, no. 4, pp. 3662–3668, 2018.

[10] Y.-H. Choi, T.-K. Lee, S.-H. Baek, and S.-Y. Oh, "Online complete coverage path planning for mobile robots based on linked spiral paths using constrained inverse distance transform," in *2009 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2009, pp. 5788–5793.

[11] P. Toth and D. Vigo, "An overview of vehicle routing problems," *The vehicle routing problem*, pp. 1–26, 2002.

[12] ——, *The vehicle routing problem*. SIAM, 2002.

[13] K. Elamvazhuthi and S. Berman, "Nonlinear generalizations of diffusion-based coverage by robotic swarms," in *2018 IEEE Conference on Decision and Control (CDC)*. IEEE, 2018, pp. 1341–1346.

[14] S. Biswal, K. Elamvazhuthi, and S. Berman, "Decentralized control of multiagent systems using local density feedback," *IEEE Transactions on Automatic Control*, vol. 67, no. 8, pp. 3920–3932, 2021.

[15] G. M. Atınç, D. M. Stipanović, and P. G. Voulgaris, "A swarm-based approach to dynamic coverage control of multi-agent systems," *Automatica*, vol. 112, p. 108637, 2020.

[16] C. Song, G. Feng, Y. Fan, and Y. Wang, "Decentralized adaptive awareness coverage control for multi-agent networks," *Automatica*, vol. 47, no. 12, pp. 2749–2756, 2011.

[17] A. Dirafzoon, M. B. Menhaj, and A. Afshar, "Decentralized coverage control for multi-agent systems with nonlinear dynamics," *IEICE TRANSACTIONS on Information and Systems*, vol. 94, no. 1, pp. 3–10, 2011.

[18] V. Krishnan and S. Martínez, "A multiscale analysis of multi-agent coverage control algorithms," *Automatica*, vol. 145, p. 110516, 2022.

[19] Y. Feng, G. Lu, W. Bai, J. Zhao, Y. Bai, and T. Xu, "Rapid coverage control with multi-agent systems based on k-means algorithm," in *2020 7th International Conference on Information, Cybernetics, and Computational Social Systems (ICCSS)*, 2020, pp. 870–874.

[20] A. A. Adepegba, "Multi-agent area coverage control using reinforcement learning techniques," Ph.D. dissertation, Université d'Ottawa/University of Ottawa, 2016.

[21] A. Dai, R. Li, Z. Zhao, and H. Zhang, "Graph convolutional multi-agent reinforcement learning for uav coverage control," in *2020 International Conference on Wireless Communications and Signal Processing (WCSP)*. IEEE, 2020, pp. 1106–1111.

[22] J. Xiao, G. Wang, Y. Zhang, and L. Cheng, "A distributed multi-agent dynamic area coverage algorithm based on reinforcement learning," *IEEE Access*, vol. 8, pp. 33 511–33 521, 2020.

[23] M. Kouzehgar, M. Meghjani, and R. Bouffanais, "Multi-agent reinforcement learning for dynamic ocean monitoring by a swarm of buoys," in *Global Oceans 2020: Singapore–US Gulf Coast*. IEEE, 2020, pp. 1–8.

[24] Y. Bai, Y. Wang, M. Svinin, E. Magid, and R. Sun, "Adaptive multi-agent coverage control with obstacle avoidance," *IEEE Control Systems Letters*, vol. 6, pp. 944–949, 2021.

[25] M. T. Nguyen, L. Rodrigues, C. S. Maniu, and S. Olaru, "Discretized optimal control approach for dynamic multi-agent decentralized coverage," in *2016 IEEE International Symposium on Intelligent Control (ISIC)*. IEEE, 2016, pp. 1–6.

[26] F. Abbasi, A. Mesbahi, and J. M. Velni, "A new voronoi-based blanket coverage control method for moving sensor networks," *IEEE Transactions on Control Systems Technology*, vol. 27, no. 1, pp. 409–417, 2017.

[27] W. Luo and K. Sycara, "Voronoi-based coverage control with connectivity maintenance for robotic sensor networks," in *2019 International Symposium on Multi-Robot and Multi-Agent Systems (MRS)*. IEEE, 2019, pp. 148–154.

[28] S. Patel, S. Hariharan, P. Dhulipala, M. C. Lin, D. Manocha, H. Xu, and M. Otte, "Multi-agent coverage in urban environments," *arXiv preprint arXiv:2008.07436*, 2020.

[29] I. E. Nielsen, D. Dera, G. Rasool, R. P. Ramachandran, and N. C. Bouaynaya, "Robust explainability: A tutorial on gradient-based attribution methods for deep neural networks," *IEEE Signal Processing Magazine*, vol. 39, no. 4, pp. 73–84, 2022.

[30] H. Rastgoftar, E. M. Atkins, and I. V. Kolmanovsky, "Scalable vehicle team continuum deformation coordination with eigen decomposition," *IEEE Transactions on Automatic Control*, vol. 67, no. 5, pp. 2514–2521, 2021.

[31] H. Rastgoftar and I. V. Kolmanovsky, "A spatio-temporal reference trajectory planner approach to collision-free continuum deformation coordination," *Automatica*, vol. 142, p. 110255, 2022.

[32] ——, "Safe affine transformation-based guidance of a large-scale multi-quadcopter system," *IEEE Transactions on Control of Network Systems*, vol. 8, no. 2, pp. 640–653, 2021.

[33] A. E. Asslouj and H. Rastgoftar, "Quadcopter tracking using euler-angle-free flatness-based control," *arXiv preprint arXiv:2212.01540*, 2022.

[34] A. El Asslouj and H. Rastgoftar, "Quadcopter tracking using euler-angle-free flatness-based control," in *2023 European Control Conference (ECC)*. IEEE, 2023, pp. 1–6.

[35] H. Rastgoftar, "Integration of a* search and classic optimal control for safe planning of continuum deformation of a multiquadcopter system," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 58, no. 5, pp. 4119–4134, 2022.

Hossein Rastgoftar an Assistant Professor at the University of Arizona. Prior to this, he was an adjunct Assistant Professor at the University of Michigan from 2020 to 2021. He was also an Assistant Research Scientist (2017 to 2020) and a Postdoctoral Researcher (2015 to 2017) in the Aerospace Engineering Department at the University of Michigan Ann Arbor. He received the B.Sc. degree in mechanical engineering-thermo-fluids from Shiraz University, Shiraz, Iran, the M.S. degrees in mechanical systems and solid mechanics from Shiraz University and the University of Central Florida, Orlando, FL, USA, and the Ph.D. degree in mechanical engineering from Drexel University, Philadelphia, in 2015.