A fixed-time formation-containment control scheme for multi-agent systems with motion planning: Applications to quadcopter UAVs

Yu-Hsiang Su, Parijat Bhowmick, *Member, IEEE*, and Alexander Lanzon, *Senior Member, IEEE*

***Abstract*—This paper proposes a fixed-time formation-containment tracking control scheme with a safe and reliable motion planning strategy. A practical scenario is considered where no agents have prior environmental knowledge, and only a subgroup of agents, defined as leaders, is equipped with obstacle-detection sensors. The primary objective is to drive agents towards the goal position while safely avoiding unknown obstacles and navigating through choke points. First, we ensure that the leader agents achieve a prescribed formation within a fixed time, and then, the follower agents converge inside a convex hull spanned by the leaders. After that, motion planning and choke point navigation algorithms are developed to achieve group obstacle avoidance and choke point navigation missions. The results demonstrate that networked agents could safely move towards the goal positions, effectively navigating unknown obstacles and manoeuvring through choke points, even when obstacle-detection sensors are limited to leader agents. Finally, the feasibility and effectiveness of the proposed control scheme were validated through an in-depth simulation case study and real-time flight experiments on a multi-UAV system.**

I. Introduction

Cooperative control of multi-agent systems (MASs) has been extensively studied and implemented over the past two decades. It aims to design control schemes for networked MASs depending on the agent's dynamics, their interaction topology, and the cooperative control objectives [1]. The coordinated control of networked unmanned aerial vehicles (UAVs) has experienced significant advancements with the progress in UAV technologies and embedded system design, as highlighted in [2] and the references therein. A multi-UAV system can be utilised to solve various real-world problems, such as localisation, mapping and navigation [3]; search and rescue operations in hazardous environments [4]; cooperative payload transportation [5], [6]; etc. Formation tracking and containment control of MASs have emerged as two prominent research topics in the last two decades. Formation tracking involves networked agents achieving a prescribed formation and tracking a moving virtual leader agent/target [7]. In contrast, containment control navigates follower agents to converge inside a convex hull spanned by the leaders [8].

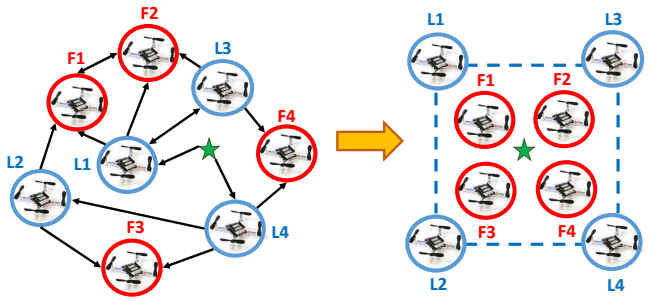


Fig. 1: Formation-containment control of a networked multi-UAV system consists of leader and follower agents. The green stars mark the virtual leader agent/target, while the black arrows represent the communication links among the agents.

Research on the design of cooperative control strategies for MASs employing the information consensus approach dates back to 2004 [9]. Recent developments in this domain have been reported in [10], [11], [12], [13], [14], [15], [16], [17], etc. However, earlier works did not guarantee fixed-time convergence, meaning that the execution of formation or containment control actions cannot be accomplished within a given fixed time. This is a significant drawback in real-world scenarios with time constraints, such as search, rescue and surveillance missions [18], where MASs are required to achieve formation or containment control within a given fixed time. Formation-containment control, as illustrated in Fig. 1, combines formation and containment control properties, offering distinct advantages for real-world applications. The article [19] studied the formation-containment control problem for second-order MASs with directed interacting topologies and implemented the control laws on multiple quadcopters. More recently, [20] proposed an adaptive formation-containment control scheme for MASs and validated it using a group of two-wheeled mobile robots. This control framework was further extended in [4] to address clustering and obstacle avoidance issues. However, these studies fell short in covering other group behaviours for MASs, such as motion planning, group obstacle avoidance, formation scaling, and choke point

This work was supported by the Engineering and Physical Sciences Research Council (EPSRC) [grant number EP/R008876/1] and by the Science and Engineering Research Board (SERB), DST, India [grant numbers SRG/2022/000892 and CRG/2022/ 006114]. All research data supporting this publication are directly available within this publication. For the purpose of open access, the authors have applied a Creative Commons Attribution (CC BY) licence to any Author Accepted Manuscript version arising.

Y.-H. Su and A. Lanzon are with the Control Systems and Robotics Group, Department of EEE, School of Engineering, University of Manchester, Manchester, M13 9PL, UK, and P. Bhowmick is with the Department of EE, IIT Guwahati, Assam - 781039, India. Emails: Yu-Hsiang.Su@manchester.ac.uk, parijat.bhowmick@iitg.ac.in, Alexander.Lanzon@manchester.ac.uk.

(i.e. a type of environmental obstruction such as windows, doors, corridors, or narrow canyons) navigation aspects.

Apart from cooperative control, cooperative motion planning plays a crucial role in MAS research, facilitating collision and obstacle avoidance. [4] exploited artificial potential field (APF) techniques to design a decentralised obstacle and collision avoidance strategy for MASs. A formation control algorithm with obstacle avoidance relying on an improved APF algorithm was reported in [21]. Furthermore, in [22], a trajectory planning methodology was proposed for quadcopters operating in a known obstacle-filled environment. However, it is worth noting that many obstacle avoidance algorithms either require each agent to be equipped with obstacle-detection sensors (e.g. [4], [21]) or rely on prior knowledge of the environment (e.g. [22]). Unless properly handled, the consensus or formation among the agents could be disrupted due to collision events among agents or between an agent and an obstacle, which could result in mission failure or damage to the agents.

Urged by the continuous need to improve cooperative control techniques for achieving fixed-time convergence, this paper develops a state-of-the-art fixed-time formation-containment tracking control scheme for MASs that also serves motion planning objectives. The focus is on a practical scenario where no agents have prior environmental knowledge and only the leader agents are equipped with obstacle-detection sensors. Under these constraints, existing motion planning and obstacle avoidance algorithms for MASs would not be practical, given that follower agents do not have any obstacle detection capabilities. Therefore, our proposed solution emphasises the integration of formation and containment control, wherein leader agents establish a safe region and guide follower agents through unknown environments. The proposed formation-containment control scheme guarantees that leader agents maintain a prescribed formation and track a moving virtual leader agent/target. At the same time, follower agents remain inside a safe region (i.e. the convex hull) guarded by the leaders. Therefore, this control scheme facilitates the safe movement of followers between locations, using the protection of the leaders. This protection enables followers to avoid unknown obstacles and navigate through choke points despite their lack of obstacle detection capabilities. In summary, the primary contributions of this paper are encapsulated in the following key points:

- A new distributed *fixed-time* cooperative control scheme is designed for networked MASs that enables agents to accomplish both formation tracking and containment control objectives within a given fixed time;
- The paper has also developed a group obstacle avoidance strategy and a choke point navigation algorithm, effectively addressing cooperative motion planning objectives even in a cluttered and obstacle-prone environment. This strategic development considers situations where only the leader agents have obstacle detection capabilities;
- The group obstacle avoidance and choke point navigation strategies together form a high-level motion planning unit, which has been integrated with the proposed fixed-time formation-containment control protocol to create a hierarchical cooperative control framework;
- As an additional contribution, the paper conducted two real-time flight experiments using a fleet of Crazyflie quadcopter UAVs to test the feasibility and performance of the proposed hierarchical cooperative control framework.

The rest of the paper is organised as follows: Section II gives essential preliminaries and describes the problem statement. Section III lays out the main theoretical developments of the paper. Section IV presents a MATLAB simulation case study that compares the fixed-time convergence property between the proposed fixed-time formation-containment tracking controller and a benchmark controller. Section V includes two real-time flight experiments conducted on Crazyflie quadcopter UAVs to validate the proposed control algorithms. Finally, Section VI summarises the paper.

The notations and acronyms are standard throughout. $\mathbb{R}$ denotes the set of all real numbers. Let $\mathbb{R}_{>0}$ and $\mathbb{R}_{\geq 0}$ denote the set of positive real numbers and the set of non-negative real numbers. Let $\mathbf{1}_n$ be the column vector with all n entries equal to 1. $\|.\|$ represents the 2-norm of a vector or a matrix.

II. Preliminaries and Problem Formulation

A. Interaction topology

A team of networked agents exchanges information according to an interaction topology. In this work, a weighted directed graph $\mathcal{G} = \{\mathcal{V}, \mathcal{E}, \mathcal{A}\}$ is used to describe the interaction topology among the agents. $\mathcal{V} = \{v_1, \ldots, v_N\}$ is the node set, $\mathcal{E} \subset \mathcal{V} \times \mathcal{V}$ is the edge set and $\mathcal{A} = [a_{ij}] \in \mathbb{R}^{N\times N}$ is the associated adjacency matrix respectively. The edge $e_{ji} = (v_j, v_i) \in \mathcal{E}$ denotes the information passes from node j to node i, which means that node j is an in-neighbour of node i. In addition, we define the set of all in-neighbours of node i as $\mathcal{N}_i = \{v_j | (v_j, v_i) \in \mathcal{E}\}$. a_{ij} represents the weight of e_{ji}. The adjacency matrix $\mathcal{A}$ is defined as $a_{ii} = 0$, $a_{ij} > 0$ if $e_{ji} \in \mathcal{E}$ and $a_{ij} = 0$ otherwise. The Laplacian matrix $\mathcal{L} = [l_{ij}] \in \mathbb{R}^{N\times N}$ associated with $\mathcal{G}$ is defined by $l_{ii} = \sum_{j\neq i} a_{ij}$ and $l_{ij} = -a_{ij}$ when $i \neq j$. A directed graph is said to have a directed spanning tree if the graph has at least one node (called the *root node*) with directed paths to every other node.

B. Problem statement

Consider a team of N networked agents, consisting of M followers and $N-M$ leaders, navigating through an unknown environment. An agent with an obstacle-detection sensor is defined as a leader, while those without are followers. We denote the sets of followers and leaders as $F = \{1, 2, \ldots, M\}$ and $L = \{M+1, M+2, \ldots, N\}$, respectively. Inspired by [11], [12], [23], a two-loop control scheme has been employed, as shown in Fig. 2. In this scheme, the inner loop applies a cascaded PID controller block such that the closed-loop translational dynamics of each quadcopter agent can be approximated by a double integrator system, as given below:

$$\begin{cases} \dot{\mathbf{p}}_i = \mathbf{v}_i, \\ \dot{\mathbf{v}}_i = \mathbf{u}_i, \end{cases} \tag{1}$$

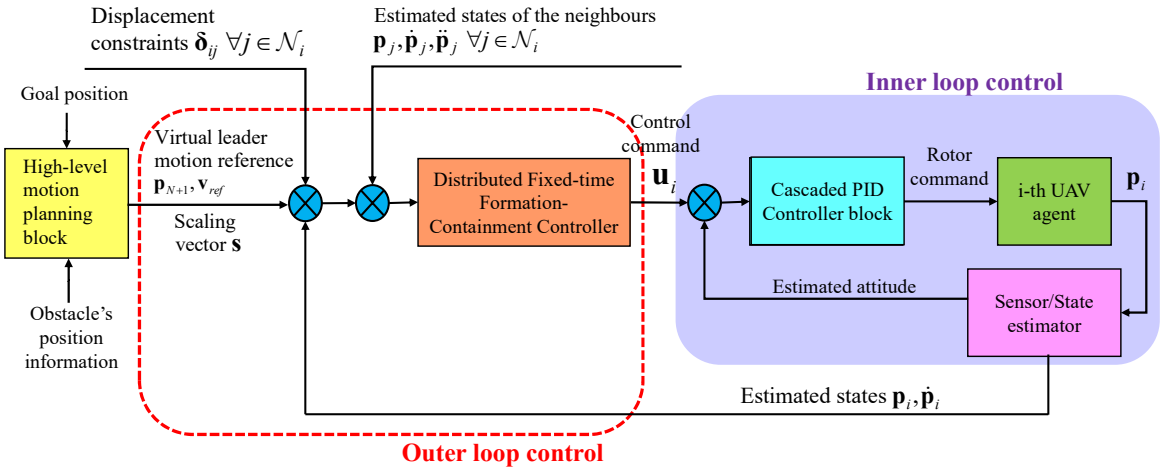


Fig. 2: The proposed hierarchical control framework. For the i^{th} leader agent, the fixed-time formation tracking controller computes the desired control command $\mathbf{u}_i$ given the motion reference of the virtual leader agent $\mathbf{p}_{N+1}$ and $\mathbf{v}_{\text{ref}}$, the scaling vector $\mathbf{s}$, its estimated states $\mathbf{p}_i$ and $\dot{\mathbf{p}}_i$, its in-neighbouring estimated states $\mathbf{p}_j$, $\dot{\mathbf{p}}_j$, $\ddot{\mathbf{p}}_j$, $\forall j \in \mathcal{N}_i$ and the formation offset $\boldsymbol{\delta}_{ij}$, $\forall j \in \mathcal{N}_i$. For the k^{th} follower agent, the fixed-time containment controller computes the desired control command $\mathbf{u}_k$ given its estimated states $\mathbf{p}_k$ and $\dot{\mathbf{p}}_k$ and its in-neighbouring estimated states $\mathbf{p}_j$, $\dot{\mathbf{p}}_j$ and $\ddot{\mathbf{p}}_j$, $\forall j \in \mathcal{N}_k$.

where $\mathbf{p}_i \in \mathbb{R}^3$ is the position vector, $\mathbf{v}_i \in \mathbb{R}^3$ is the velocity vector, and $\mathbf{u}_i \in \mathbb{R}^3$ is the control input vector (i.e. the desired acceleration) for the i^{th} agent respectively. In addition, we introduce a virtual leader agent/target, labelled as the $(N+1)^{\text{th}}$ node, to provide a reference trajectory for all networked agents to track. This virtual leader agent/target is treated as an exo-system, independent of the agent dynamics. The formation configuration vector for each agent w.r.t. this virtual leader agent/target is defined as $\mathbf{h}_i \in \mathbb{R}^3$, $\forall i \in L$, which is determined by displacement constraints specified by users. Let $\boldsymbol{\delta}_{ij} \in \mathbb{R}^3$ denote the formation offset between the i^{th} and j^{th} agents, where $i \in L$, $j \in \mathcal{N}_i$. The relation between the formation configuration and formation offset is given by $\boldsymbol{\delta}_{ij} = \mathbf{h}_i - \mathbf{h}_j$. A directed graph $\mathcal{G}$ describes the interaction topology among each agent. We will now declare an assumption on the interaction topology and an essential technical result.

Assumption 1: The interaction topology contains a directed spanning tree with the virtual leader agent being the root node. Each leader's in-neighbours are only the virtual leader agent and leaders. For each follower, at least one leader has a directed path to that follower.

According to Assumption 1, we can partition the Laplacian matrix $\mathcal{L}$ into the following form:

$$\mathcal{L} = \begin{bmatrix} \mathcal{L}_{ff} & \mathcal{L}_{fl} & 0_{M\times 1} \\ 0_{(N-M)\times M} & \mathcal{L}_{ll} & \mathcal{L}_{lv} \\ 0_{1\times M} & 0_{1\times(N-M)} & 0_{1\times 1} \end{bmatrix}. \tag{2}$$

Where $\mathcal{L}_{ff} \in \mathbb{R}^{M\times M}$ is the sub-Laplacian matrix that represents the interactions among the followers, $\mathcal{L}_{fl} \in^{M\times(N-M)}$ is the sub-Laplacian matrix that describes the relationship among the followers and leaders, $\mathcal{L}_{ll} \in \mathbb{R}^{(N-M)\times(N-M)}$ is the sub-Laplacian matrix corresponding to the leaders and lastly, $\mathcal{L}_{lv} \in \mathbb{R}^{(N-M)\times 1}$ is the sub-Laplacian matrix that indicates the relationship among the leaders and the virtual leader agent.

Lemma 1: [24] Each entry of $-\mathcal{L}_{ll}^{-1}\mathcal{L}_{lv}$ is non-negative and each row sum of $-\mathcal{L}_{ll}^{-1}\mathcal{L}_{lv}$ is equal to one.

The primary aim of this work is to formulate a control framework suitable for a practical scenario wherein no agents have prior environmental knowledge, and only leader agents are equipped with obstacle-detection sensors. This framework is designed to achieve the following objectives: (i) ensuring that all leader agents achieve and maintain formation control within a given fixed time while tracking the motion reference of the virtual leader agent; (ii) guaranteeing that all follower agents achieve and maintain containment control within a given fixed time while tracking the motion reference of the virtual leader agent; and (iii) ensuring that all networked agents safely avoid unknown obstacles, navigate through choke points and ultimately reach the goal position.

III. Main Theoretical Development

This section presents the key theoretical developments of this paper. As shown in Fig. 2, we propose a hierarchical control framework for MASs to solve the challenges of fixed-time formation-containment tracking control and cooperative motion planning.

Given that only leader agents are equipped with obstacle-detection sensors, we employ a formation-containment technique to govern the motions of networked agents. The primary responsibility of leader agents is to establish and maintain a safe region for follower agents, which do not have any obstacle detection capabilities. This strategy allows follower agents to safely navigate around unknown obstacles while moving toward the goal position, provided they remain inside this safe region. In particular, leader agents must maintain a formation configuration, even during navigation around unknown obstacles, as this configuration serves as a safe region for follower agents. We must note that we approach obstacle avoidance as a collective behaviour for MASs instead of an individual

behaviour for each agent. This approach distinguishes our work from most existing methods.

A. A hierarchical cooperative control framework

There are three levels in the proposed hierarchical control framework shown in Fig. 2. First, the virtual leader agent/target implements a high-level motion planning algorithm. It is followed by a two-loop control scheme, wherein each agent computes its desired control command using the outer loop distributed fixed-time formation-containment tracking controller. This wanted control command is then tracked through an inner loop PID controller.

B. Distributed formation-containment control of MASs

We propose a distributed formation-containment tracking controller to achieve formation-containment and tracking behaviours for networked agents whose closed-loop translational dynamics are approximated by (1).

We will now give the definitions of formation and containment control. For the leader agents, the formation tracking error $\boldsymbol{\xi}_i$ w.r.t. its in-neighbours is defined as:

$$\boldsymbol{\xi}_i = \sum_{j\in\mathcal{N}_i} a_{ij}(\mathbf{p}_i - \mathbf{p}_j - \boldsymbol{\delta}_{ij}) \quad \forall i \in L. \tag{3}$$

For the follower agents, the containment error $\boldsymbol{\xi}_k$ w.r.t. its in-neighbours is defined as:

$$\boldsymbol{\xi}_k = \sum_{j\in\mathcal{N}_k} a_{kj}(\mathbf{p}_k - \mathbf{p}_j) \quad \forall k \in F. \tag{4}$$

In addition, we define the velocity errors $\boldsymbol{\zeta}_i$ and $\boldsymbol{\zeta}_k$ for each leader and follower agent w.r.t. its in-neighbours as

$$\boldsymbol{\zeta}_i = \sum_{j\in\mathcal{N}_i} a_{ij}(\mathbf{v}_i - \mathbf{v}_j) \quad \forall i \in L, \tag{5}$$

and

$$\boldsymbol{\zeta}_k = \sum_{j\in\mathcal{N}_k} a_{kj}(\mathbf{v}_k - \mathbf{v}_j) \quad \forall k \in F. \tag{6}$$

Definition 1: (**Formation control**) The leaders are said to achieve the prescribed formation, specified by the formation configuration vector $\mathbf{h}$ w.r.t. the virtual leader agent/target, if the position of each leader agent $\mathbf{p}_i(t)$ satisfies the relationship

$$\lim_{t\to\infty} \mathbf{p}_i(t) - \mathbf{p}_j(t) = \boldsymbol{\delta}_{ij} \tag{7}$$

for $i \in L$ and $j \in \mathcal{N}_i$, where

$$\boldsymbol{\delta}_{ij} = \mathbf{h}_i - \mathbf{h}_j \qquad \forall j \in \mathcal{N}_i. \tag{8}$$

Lemma 2: Consider a MAS connected via a directed graph that satisfies Assumption 1. The leader agents are said to achieve the prescribed formation, specified by $\mathbf{h}$ and satisfying (8), if and only if

$$\lim_{t\to\infty}\left(\sum_{j\in\mathcal{N}_i} a_{ij}\Big(\mathbf{p}_i(t)-\mathbf{p}_j(t)-\boldsymbol{\delta}_{ij}\Big)\right) = \lim_{t\to\infty}\boldsymbol{\xi}_i(t) = 0 \tag{9}$$

for all $i \in L$.

Proof: Since the interaction topology satisfies Assumption 1, we have $\boldsymbol{\delta}_{ij} = \mathbf{h}_i - \mathbf{h}_j$ as mentioned in (8). Note that $\mathbf{h}_i, \mathbf{h}_j \in \mathbb{R}^3$ for all i and j. Then, we can express the formation tracking error $\boldsymbol{\xi}_i$, given in (3), as

$$\boldsymbol{\xi}_i = \sum_{j=M+1}^{N} a_{ij}\Big((\mathbf{p}_i - \mathbf{h}_i) - (\mathbf{p}_j - \mathbf{h}_j)\Big) + a_{i,N+1}\Big((\mathbf{p}_i - \mathbf{h}_i) - \mathbf{p}_{N+1}\Big). \tag{10}$$

Define $\mathbf{z}_i = \mathbf{p}_i - \mathbf{h}_i$, $\mathbf{p} = [\mathbf{p}_{M+1}^\top, \mathbf{p}_{M+2}^\top, \ldots, \mathbf{p}_N^\top]^\top$, $\mathbf{z} = [\mathbf{z}_{M+1}^\top, \mathbf{z}_{M+2}^\top, \ldots, \mathbf{z}_N^\top]^\top$, $\mathbf{h} = [\mathbf{h}_{M+1}^\top, \mathbf{h}_{M+2}^\top, \ldots, \mathbf{h}_N^\top]^\top$ and $\boldsymbol{\xi} = [\boldsymbol{\xi}_{M+1}^\top, \boldsymbol{\xi}_{M+2}^\top, \ldots, \boldsymbol{\xi}_N^\top]^\top$. We can express $\boldsymbol{\xi}$ in the Kronecker product form as

$$\boldsymbol{\xi} = (\mathcal{L}_{ll} \otimes I_3)\mathbf{z} + (\mathcal{L}_{lv} \otimes I_3)\mathbf{p}_{N+1}. \tag{11}$$

Now, if (9) holds, we have $\lim_{t\to\infty} \boldsymbol{\xi}(t) = 0$, which in turn implies from (11)

$$\lim_{t\to\infty}\Big[\mathbf{p}(t) - \mathbf{h}(t) + \left(\mathcal{L}_{ll}^{-1}\mathcal{L}_{lv} \otimes I_3\right)\mathbf{p}_{N+1}(t)\Big] = 0. \tag{12}$$

According to Assumption 1 and Lemma 1, we have $\mathcal{L}_{ll}^{-1}\mathcal{L}_{lv} = -\mathbf{1}_{N-M}$. Substituting $\mathcal{L}_{ll}^{-1}\mathcal{L}_{lv} = -\mathbf{1}_{N-M}$ into (12), we get

$$\lim_{t\to\infty}\Big[\mathbf{p}(t) - \mathbf{h}(t) - \mathbf{1}_{N-M} \otimes I_3\mathbf{p}_{N+1}(t)\Big] = 0. \tag{13}$$

From equation (13), we can conclude that the prescribed formation among the leader agents satisfies the condition in (7). This demonstrates that the prescribed formation is achieved if (9) holds. ■

Remark 1: Formation control typically refers to scenarios in which leader agents achieve the prescribed formation w.r.t. a fixed or static virtual leader agent. On the other hand, formation tracking control involves leader agents aiming to achieve the prescribed formation w.r.t. a moving or dynamic virtual leader agent.

Definition 2: (**Containment control**) The followers are said to achieve containment if each follower position $\mathbf{p}_k(t)$ asymptotically converges to the convex hull spanned by those leader agents. This is equivalent to the condition that for any $k \in F$ and $j \in L$, there exist non-negative constants α_{kj} satisfying $\sum_{j=M+1}^{N} \alpha_{kj} = 1$ such that

$$\lim_{t\to\infty}\left(\mathbf{p}_k(t) - \sum_{j\in L}\alpha_{kj}\mathbf{p}_j(t)\right) = 0. \tag{14}$$

We are ready to present the benchmark theorem, establishing a formation-containment tracking control for networked agents modelled by double-integrator dynamics.

Theorem 1: Consider N agents connected via a directed graph $\mathcal{G}$ that satisfies Assumption 1. The closed-loop translational dynamics of each agent are approximated by double integrator dynamics, as given in equation (1). Then, the MAS achieves the formation-containment tracking control under the action of the distributed control protocol

$$\begin{cases} \mathbf{u}_i = -\dfrac{1}{\gamma_i}\Big(k_p\boldsymbol{\xi}_i + k_v\boldsymbol{\zeta}_i - \displaystyle\sum_{j\in\mathcal{N}_i} a_{ij}\dot{\mathbf{v}}_j\Big) \quad \forall i \in L, \\ \mathbf{u}_k = -\dfrac{1}{\gamma_k}\Big(k_p\boldsymbol{\xi}_k + k_v\boldsymbol{\zeta}_k - \displaystyle\sum_{j\in\mathcal{N}_k} a_{kj}\dot{\mathbf{v}}_j\Big) \quad \forall k \in F, \end{cases} \tag{15}$$

where $\gamma_i = \sum_{j\in\mathcal{N}_i} a_{ij}$, $\gamma_k = \sum_{j\in\mathcal{N}_k} a_{kj}$, $k_p > 0$, $k_v > 0$ and $\dot{\mathbf{v}}_j$ is the acceleration vector of in-neighbouring agents.

Proof: The closed-loop translational dynamics of each agent in equation (1) are decoupled along the X, Y and Z axes, which means that the analysis on the one-dimensional case is valid for three-dimensional motions [25]. Therefore, without loss of generality and for brevity, we can assume p_i, v_i, $u_i \in \mathbb{R}^1$ in the rest of this proof.

First, we obtain the closed-loop dynamics for leaders by plugging the control law (15) into (1), yielding

$$\dot{v}_i = -\frac{1}{\gamma_i}\Big(k_p\xi_i + k_v\zeta_i - \sum_{j\in\mathcal{N}_i} a_{ij}\dot{v}_j\Big) \quad \forall i \in L. \tag{16}$$

Multiplying γ_i on both sides of (16) gives

$$\sum_{j\in\mathcal{N}_i} a_{ij}(\dot{v}_i - \dot{v}_j) = -k_p\xi_i - k_v\zeta_i. \tag{17}$$

By differentiating ζ_i, and the facts that $\dot{\zeta}_i = \ddot{\xi}_i$ and $\zeta_i = \dot{\xi}_i$, we get the closed-loop formation tracking error dynamics as

$$\ddot{\xi}_i = -k_p\xi_i - k_v\dot{\xi}_i. \tag{18}$$

Since $k_p > 0$ and $k_v > 0$, it follows both ξ_i and $\dot{\xi}_i = \zeta_i$ asymptotically converge to zero for all $i \in L$. We conclude that all leader agents achieve and maintain the prescribed formation configuration while tracking the virtual leader agent.

Following the method as mentioned above, we can get the closed-loop containment error dynamics for each follower agent as

$$\ddot{\xi}_k = -k_p\xi_k - k_v\dot{\xi}_k. \tag{19}$$

Since $k_p > 0$ and $k_v > 0$, it follows both ξ_k and $\dot{\xi}_k = \zeta_k$ asymptotically converge to zero for all $k \in F$. Thus, we conclude that all follower agents converge and remain inside the convex hull spanned by those leaders while tracking the virtual leader agent.

As a result, we prove that the formation-containment tracking problem for networked agents whose closed-loop translational dynamics are approximated by double-integrator models is solved via the proposed distributed control protocol (15). This completes the proof. ∎

Remark 2: t is worth noting that the need for acceleration feedback from the in-neighbouring agents may lead to an implementation problem. In practice, the agents can transmit their acceleration data (measured by an onboard accelerometer) to their in-neighbours through wireless or Bluetooth communication. A similar technique was used in [26], [27]. Otherwise, the acceleration data can be estimated using a Kalman Filter, EKF or UKF algorithm. Another option may also be possible that the acceleration data can be generated from the measured velocity data via numerical differentiation (e.g. forward/backward difference method), and the inaccuracies or noises can be removed by designing appropriate filters.

C. Fixed-time formation-containment control of MASs

In Theorem 1, the convergence of formation tracking and containment errors is only guaranteed at an infinite time horizon. However, practical applications often require that networked agents achieve formation-containment and tracking behaviours within a given fixed time. The formation tracking and containment errors must converge to zero within a fixed time. Inspired by this practical constraint, we will extend Theorem 1 to propose a novel fixed-time formation-containment tracking controller for MASs.

Motivated by [28], [29], we introduce a time-varying function

$$\mu(t) = \begin{cases} \left(\frac{T}{T-t}\right)^h & \forall t < T, \\ 0 & \forall t \geq T, \end{cases} \tag{20}$$

where T is a fixed convergence time, and h is a user-defined parameter. We will now introduce two important lemmas that will be used to prove the fixed-time convergence theorem.

Lemma 3: [30] If x and y are non-negative real numbers and p and q are positive real numbers such that $\frac{1}{p} + \frac{1}{q} = 1$, then $xy \leq \frac{x^p}{p} + \frac{y^q}{q}$. The equality condition holds if and only if $x^p = y^q$.

Lemma 4: [29] Consider a continuously differentiable function $f : \mathbb{R} \to \mathbb{R}_{\geq 0}$, and its derivative satisfies

$$\dot{f}(t) \leq -\alpha f - \beta\frac{\dot{\mu}}{\mu}f \quad \forall t \geq 0, \tag{21}$$

where $\alpha \in \mathbb{R}_{>0}$ and $\beta \in \mathbb{R}_{>0}$. Then, we have

$$f(t) \begin{cases} \leq \mu^{-\beta}e^{-\alpha t}f(0) & \forall t < T, \\ = 0 & \forall t \geq T. \end{cases} \tag{22}$$

We are now ready to extend Theorem 1 to propose the fixed-time formation-containment tracking control theorem for networked agents modelled by double-integrator dynamics.

Theorem 2: Consider N agents connected via a directed graph $\mathcal{G}$ that satisfies Assumption 1. The closed-loop translational dynamics of each agent are approximated by double integrator dynamics, as given in equation (1). Then, the MAS achieves the formation-containment and tracking control within a given fixed time T under the action of the distributed fixed time control protocol

$$\begin{cases} \mathbf{u}_i = -\frac{1}{\gamma_i}\Big((\alpha + \beta\frac{\dot{\mu}}{\mu})\boldsymbol{\xi}_i + (\alpha + \beta\frac{\dot{\mu}}{\mu})\boldsymbol{\zeta}_i - \sum_{j\in\mathcal{N}_i} a_{ij}\dot{\mathbf{v}}_j\Big) \\ \qquad\qquad\qquad\qquad\qquad\qquad \forall i \in L, \\ \mathbf{u}_k = -\frac{1}{\gamma_k}\Big((\alpha + \beta\frac{\dot{\mu}}{\mu})\boldsymbol{\xi}_k + (\alpha + \beta\frac{\dot{\mu}}{\mu})\boldsymbol{\zeta}_k - \sum_{j\in\mathcal{N}_k} a_{kj}\dot{\mathbf{v}}_j\Big) \\ \qquad\qquad\qquad\qquad\qquad\qquad \forall k \in F, \end{cases} \tag{23}$$

where $\alpha > 0$, $\beta > 0$ are two control gains to be designed.

Proof: Following a similar procedure as in Theorem 1, we can obtain the closed-loop formation tracking error dynamics for leader agents, which are

$$\ddot{\xi}_i = -(\alpha + \beta\frac{\dot{\mu}}{\mu})(\xi_i + \dot{\xi}_i). \tag{24}$$

We consider the following Lyapunov candidate function:

$$V = c\sum_{i=M+1}^{N}\xi_i^2 + \sum_{i=M+1}^{N}\dot{\xi}_i^2 \geq 0. \tag{25}$$

Its time derivative along the trajectory of (24) is

$$\begin{aligned}\dot{V} &= 2c\sum_{i=M+1}^{N}\xi_i\dot{\xi}_i + 2\sum_{i=M+1}^{N}\dot{\xi}_i\ddot{\xi}_i \\ &= 2c\sum_{i=M+1}^{N}\xi_i\dot{\xi}_i + 2\sum_{i=M+1}^{N}\dot{\xi}_i(\alpha+\beta\frac{\dot{\mu}}{\mu})(-\xi_i-\dot{\xi}_i) \\ &\leq \sum_{i=M+1}^{N} c\xi_i^2 + c\dot{\xi}_i^2 + (\alpha+\beta\frac{\dot{\mu}}{\mu})(-\xi_i^2-3\dot{\xi}_i^2) \\ &= \sum_{i=M+1}^{N}\Big(c-(\alpha+\beta\frac{\dot{\mu}}{\mu})\Big)\xi_i^2 + \Big(c-3(\alpha+\beta\frac{\dot{\mu}}{\mu})\Big)\dot{\xi}_i^2 \\ &\leq (c-\alpha)V - \beta\frac{\dot{\mu}}{\mu}V,\end{aligned} \tag{26}$$

where we have used Lemma 3 to get the inequality in (26). By choosing $c<\alpha$, the inequality in (26) can be expressed as follows

$$\dot{V} \leq -\hat{\alpha}V - \beta\frac{\dot{\mu}}{\mu}V, \tag{27}$$

where $\hat{\alpha}=\alpha-c>0$. According to Lemma 4, we have

$$\|\xi_i(t)\|\begin{cases}\leq \mu^{-\beta}e^{-\hat{\alpha}t}\|\xi_i(0)\| & \forall t<T,\\ =0 & \forall t\geq T,\end{cases}\quad \forall i\in L, \tag{28}$$

and

$$\|\zeta_i(t)\|\begin{cases}\leq \mu^{-\beta}e^{-\hat{\alpha}t}\|\zeta_i(0)\| & \forall t<T,\\ =0 & \forall t\geq T,\end{cases}\quad \forall i\in L. \tag{29}$$

That is, $\xi_i\to 0$ and $\zeta_i\to 0$ within a given fixed time T for all leader agents.

Following the method as mentioned above, we can derive that $\xi_k\to 0$ and $\zeta_k\to 0$ within a given fixed time T for all follower agents. Therefore, networked agents achieve formation-containment and tracking behaviours within a given fixed time T via the distributed fixed-time control protocol (23). This completes the proof. ∎

Remark 3: In [29], the primary focus of the research was on developing a fixed-time formation control scheme for a fixed or static virtual leader agent/target. In contrast, our study encompasses a broader scope, including fixed-time formation tracking and containment control w.r.t. a moving or dynamic virtual leader agent/target.

D. Motion planning algorithm

The primary objective of the high-level motion planning algorithm, which is at the first level of the hierarchical control framework depicted in Fig. 2, is to formulate the motion reference for the virtual leader agent/target. This motion reference should enable networked agents to safely navigate unknown environments and reach the goal position by effectively tracking the virtual leader agent. It is assumed that the virtual leader agent follows a velocity trajectory, that is $\dot{\mathbf{p}}_{N+1}=\mathbf{v}_{\text{ref}}$, where $\mathbf{p}_{N+1}\in\mathbb{R}^3$ and $\mathbf{v}_{\text{ref}}\in\mathbb{R}^3$ are the position and the motion reference vectors (i.e. the desired velocity computed from the high-level motion planning algorithm) of the virtual leader agent. In particular, we consider three group behaviours, which are: (i) goal position seeking; (ii) group obstacle avoidance; and (iii) choke point navigation.

1) Goal position seeking: Let $\mathbf{p}_g\in\mathbb{R}^3$ denote the fixed goal position vector for the virtual leader agent and $d(\mathbf{p}_{N+1})=\|\mathbf{p}_{N+1}-\mathbf{p}_g\|$. We consider the attractive APF

$$U_a(\mathbf{p}_{N+1})=\begin{cases}\frac{k_a}{2}d^2(\mathbf{p}_{N+1}) & \text{if } d(\mathbf{p}_{N+1})\leq d_g^*,\\ d_g^*k_a d(\mathbf{p}_{N+1})-\frac{k_a}{2}(d_g^*)^2 & \text{if } d(\mathbf{p}_{N+1})>d_g^*,\end{cases} \tag{30}$$

where $k_a>0$ is the attractive gain, and d_g^* is a user-defined threshold distance which is close to the goal position. The gradient of the attractive APF in (30) can be derived as

$$\nabla U_a(\mathbf{p}_{N+1})=\begin{cases}k_a(\mathbf{p}_{N+1}-\mathbf{p}_g) & \text{if } d(\mathbf{p}_{N+1})\leq d_g^*,\\ \frac{d_g^*k_a(\mathbf{p}_{N+1}-\mathbf{p}_g)}{d(\mathbf{p}_{N+1})} & \text{if } d(\mathbf{p}_{N+1})>d_g^*.\end{cases} \tag{31}$$

Next, we apply the gradient descent method and set the motion reference of the virtual leader agent as the negative gradient of the attractive APF in (31), which gives

$$\begin{aligned}\dot{\mathbf{p}}_{N+1} &= \mathbf{v}_{\text{ref}} \\ &= -\nabla U_a(\mathbf{p}_{N+1}).\end{aligned} \tag{32}$$

Applying the gradient descent method, we conclude that the virtual leader agent converges to the goal position. Then, it follows that all networked agents will reach the goal position as long as they continue tracking the motion reference of the virtual leader agent via Theorem 2.

2) Group obstacle avoidance: When networked agents autonomously navigate in unknown environments while tracking the motion reference of the virtual leader agent in (32), they often encounter obstacles. Therefore, obstacle avoidance is paramount in the cooperative motion planning problem for MASs. This subsection presents a novel group obstacle avoidance algorithm that employs the repulsive APF and curl-free vector field. The primary objective is to establish a virtual safe region centred on the virtual leader agent, guaranteeing freedom from collisions with any obstacles. Therefore, as long as the size of the formation configuration remains within this virtual safe region, all networked agents can effectively navigate around obstacles by continuously tracking the motion reference of the virtual leader agent via Theorem 2. This approach remains applicable even in scenarios where follower agents do not have any obstacle detection capabilities.

We design a virtual safe region with a three-dimensional sphere or circle shape in two sizes. This safe region has a radius of r_s and is centred at the position of the virtual leader agent. When leader agents detect an obstacle, they send this information to the high-level motion planner. Then, the motion planning algorithm generates a repulsive force to push away the virtual leader agent and the corresponding virtual safe region.

Let $\mathbf{p}_o\in\mathbb{R}^3$ denote the position vector of an obstacle and $D(\mathbf{p}_{N+1})=\|\mathbf{p}_{N+1}-\mathbf{p}_o\|$. We consider the repulsive APF

$$U_b(\mathbf{p}_{N+1})=\begin{cases}\frac{k_b}{2}(\frac{1}{D(\mathbf{p}_{N+1})}-\frac{1}{Q^*})^2 & \text{if } D(\mathbf{p}_{N+1})\leq Q^*,\\ 0 & \text{if } D(\mathbf{p}_{N+1})>Q^*,\end{cases} \tag{33}$$

$k_b > 0$ is the repulsive gain, and Q^* is a user-defined threshold distance activating the repulsive APF. Note that $Q^* > r_s$. The gradient of the above repulsive APF in (33) can be derived as

$$\nabla U_b(\mathbf{p}_{N+1}) = \begin{cases} f(\mathbf{p}_{N+1}) \frac{k_b \nabla D(\mathbf{p}_{N+1})}{D^2(\mathbf{p}_{N+1})} & \text{if } D(\mathbf{p}_{N+1}) \leq Q^*, \\ 0 & \text{if } D(\mathbf{p}_{N+1}) > Q^*, \end{cases} \tag{34}$$

where $f(\mathbf{p}_{N+1}) = \big(1/Q^* - 1/D(\mathbf{p}_{N+1})\big)$. Next, we add the negative gradient of the repulsive APF in (34) into the motion reference of the virtual leader agent in (32), which gives

$$\begin{aligned} \dot{\mathbf{p}}_{N+1} &= \mathbf{v}_{\text{ref}} \\ &= -\nabla U_a(\mathbf{p}_{N+1}) - \nabla U_b(\mathbf{p}_{N+1}). \end{aligned} \tag{35}$$

One drawback of the conventional APF techniques is the occurrence of the local minimum problem. This situation can occur when the magnitudes of the attractive and repulsive forces are equal, but their directions are opposite (i.e. $\mathbf{v}_{\text{ref}} \simeq 0$ in (35)). This can cause all agents to become trapped near obstacles and cease moving. Several research studies have addressed this issue and proposed various algorithms. These include the improved APF method [21], the stochastic method [31], and the unstable equilibrium method [32]. In this work, we have implemented the curl-free vector field method, inspired by [33], [34], to tackle the local minimum problem.

To address the local minimum problem, we employ a rotation matrix, denoted as R, to rotate the repulsive APF in (33). This rotation generates the curl-free vector field around an obstacle, denoted as $U_c(\mathbf{p}_{N+1})$. The direction of R is determined by the angle of the relative position vector between the obstacle and the virtual leader agent, denoted as ρ. The relation between the gradient of the curl-free vector field and the repulsive APF can be expressed as

$$\nabla U_c(\mathbf{p}_{N+1}) = R \nabla U_b(\mathbf{p}_{N+1}), \tag{36}$$

where

$$R = \begin{cases} \begin{bmatrix} 0 & 1 & 0 \\ -1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} & \text{if } \rho < 0 \Rightarrow \text{clockwise}, \\ \begin{bmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} & \text{if } \rho \geq 0 \Rightarrow \text{counterclockwise}. \end{cases} \tag{37}$$

Then, the new motion reference of the virtual leader agent can be represented by

$$\begin{aligned} \dot{\mathbf{p}}_{N+1} &= \mathbf{v}_{\text{ref}} \\ &= -\nabla U_a(\mathbf{p}_{N+1}) - \nabla U_c(\mathbf{p}_{N+1}). \end{aligned} \tag{38}$$

We set a user-defined threshold (i.e. $P^* \simeq 0$) to activate the curl-free vector field when the virtual leader agent becomes trapped in a local minimum in (35). Beyond this threshold, we revert to the conventional repulsive potential APF in (33) to compute the motion reference for the virtual leader agent. The above high-level motion planning method is summarised in Algorithm 1.

Assumption 2: The position of the virtual leader agent is set at the centre of the formation configuration, which is also the centre of the virtual safe region. This virtual safe region, with a user-specified radius of r_s, covers the entire formation configuration spanned by all leader agents.

Algorithm 1: A high-level motion planning algorithm for the virtual leader agent.

```
initialization;
while true do
    if D(p_{N+1}) ≤ Q* then
        set v_ref = equation (35);
        if ||v_ref|| ≤ P* then
            set v_ref = equation (38);
    else
        set v_ref = equation (32);
    end
end
```

We are now ready to integrate the distributed fixed-time formation-containment tracking controller (23) with the high-level motion planning algorithm described in Algorithm 1 to address the objectives mentioned in Section II-B. Algorithm 1 generates a motion reference for the virtual leader agent/target to reach the goal position while ensuring that a virtual safe region is collision-free from obstacles. In addition, all networked agents maintain formation-containment behaviour while tracking the motion reference of the virtual leader agent through the distributed fixed-time formation-containment tracking controller (23). Therefore, if Assumption 2 holds, all networked agents will safely reach the goal position while effectively navigating around obstacles by continuously tracking the motion reference of the virtual leader agent/target via Theorem 2.

Remark 4: Most existing works have considered obstacle avoidance as an individual behaviour for each agent in the formation control problem of MASs. Therefore, in such scenarios, networked agents may not consistently maintain their formation configuration when encountering and navigating around obstacles (e.g. [4], [21], [22], [33]). In contrast, this paper highlights the importance of leader agents maintaining the formation configuration even during obstacle navigation. This new approach allows leader agents to effectively guide follower agents who do not have any obstacle detection capabilities in avoiding unknown obstacles.

3) Choke point navigation: Next, we consider a particular navigation problem for MASs called choke point navigation. When a team of networked agents encounters a choke point, the motion reference of the virtual leader agent, as determined by equation (35), can become trapped in a local minimum of the APFs, thereby preventing all agents from passing through the choke point. To tackle this issue effectively, we introduce a scaling vector to reduce the size of the formation offset for leader agents and propose an algorithm that enables networked agents to navigate safely through choke points. This algorithm ensures that the formation offset of the leaders is smaller than the open space of the choke point, thus allowing networked agents to pass through the choke point safely.

First, we define a scaling vector $\mathbf{s} = [s_x, s_y, s_z]^\top$, where s_x, s_y and s_z are the scaling variables in the X, Y, and Z directions

respectively. The scaled formation offset can be expressed as

$$\hat{\boldsymbol{\delta}}_{ij} = [s_x\delta_{ij}^x, s_y\delta_{ij}^y, s_z\delta_{ij}^z]^\top. \tag{39}$$

When the leader agents on both sides of the virtual leader agent detect obstacles blocking the group's pathway, the motion reference of the virtual leader agent, as determined by equation (35), approaches zero, causing all networked agents to stop moving. However, suppose the leader agent in front of the virtual leader agent detects no obstacles. In that case, it signifies that a clear pathway exists from the virtual leader agent's position to the goal position, and these obstacles can be classified as a choke point. To effectively navigate through the choke point, the high-level motion planning algorithm stops the entire team's movement. Then, it gradually decreases the scaling vector in each iteration, causing all leader agents to reduce the size of the formation offset until the choke point is no longer detected in front of any leader agent. This strategy ensures the safe navigation of all networked agents through the choke point, allowing them to resume their movement towards the goal position by tracking the motion reference of the virtual leader agent in Algorithm 1. The above method is summarised in Algorithm 2.

Algorithm 2: A choke point navigation algorithm for networked agents.

```
initialization;
while true do
    if detect choke point = true then
        set v_ref = 0;
        decrease the scaling vector s;
        set δ̂_ij = equation (39);
    else
        switch to Algorithm 1 to compute v_ref;
    end
end
```

IV. MATLAB Simulation Results

In this section, a MATLAB simulation case study is provided to compare the performance of the fixed-time convergence property between the fixed-time formation-containment tracking controller in Theorem 2 and the benchmark controller in Theorem 1. We considered a 3D formation-containment control problem involving 16 agents, consisting of eight leaders and eight followers. Fig. 3 describes the interaction topology among the leader and follower agents. The primary control objective was to achieve a 3D cubic formation with the leaders while ensuring that the followers converged inside a convex hull spanned by the leaders within a given fixed time of $T = 20\,\mathrm{s}$. The initial position and velocity reference of the virtual leader agent were provided as $\mathbf{p}_{N+1} = [0, 0, 1]^\top$ and $\mathbf{v}_{\text{ref}} = [0.2, 0, 0]^\top$, respectively.

Fig. 4 compares the formation tracking errors of the leaders and containment errors of the followers, simulated by applying respectively the proposed fixed-time control protocol (23) in Theorem 2 and the benchmark control protocol (15) in

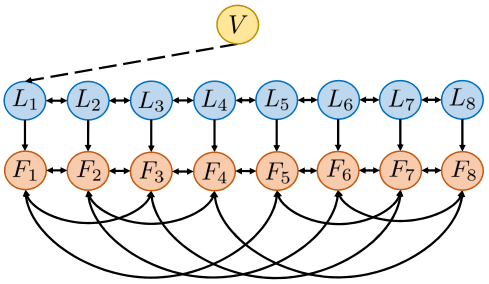
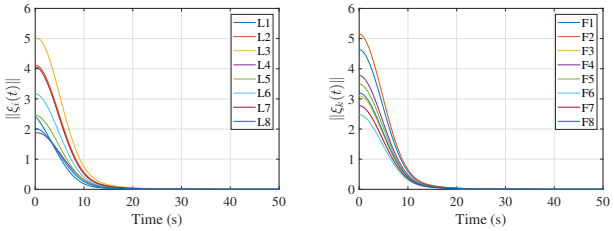
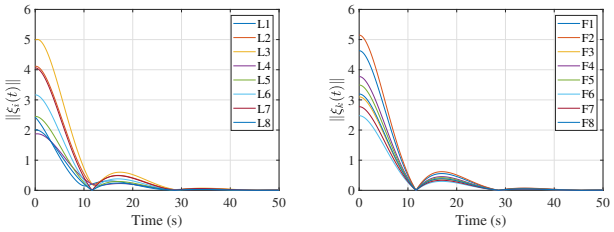


Fig. 3: Interaction topology for the simulation case study. Agents F_1 to F_8 are the followers, agents L_1 to L_8 are the leaders and agent V is the virtual leader agent.



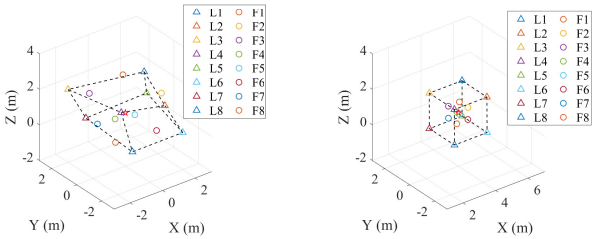
(a) Fixed-time control protocol (23) in Theorem 2. The given fixed time was $T = 20\,\mathrm{s}$, and the other parameters were $\alpha = 0.05$, $\beta = 0.4$ and $h = 5$.



(b) Benchmark control protocol (15) in Theorem 1. The control gains k_p and k_v were chosen as 0.05.

Fig. 4: [Performance comparison study:] Comparison of the i) Formation tracking error $\|\boldsymbol{\xi}_i(t)\|$, $\forall i \in L$ and ii) Containment error $\|\boldsymbol{\xi}_k(t)\|$, $\forall k \in F$, simulated by applying (a) the fixed-time control protocol; and (b) the benchmark control protocol.

Theorem 1. In the case of the fixed-time control protocol, the formation tracking and containment errors smoothly converged to zero within 20 seconds only, as shown in Fig. 4(a). However, in the case of the benchmark control protocol, the errors converged to zero after 30 seconds with some inaccuracies and oscillations during the transient response, as reported in Fig. 4(b). Fig. 5(a) shows the initial positions of the agents, where the leaders formed an irregular formation, and followers were outside the convex hull spanned by the leaders. By implementing the fixed-time formation-containment tracking controller, all leader agents achieved the prescribed 3D cubic formation, and all follower agents converged inside the convex hull spanned by those leaders within 20 seconds (i.e. the given fixed time), as shown in Fig. 5(b). The comparative study reveals that Theorem 2 solves the fixed-time formation-containment tracking problem. In addition, it provides a better



(a) $t = 0\,\mathrm{s}$ (b) $t = 20\,\mathrm{s}$

Fig. 5: [Results for Simulation case study:] The black dashed lines represent the formations of leader agents, and the red stars mark the positions of the virtual leader agent. (a) At $t = 0\,\mathrm{s}$, the leaders started from random initial locations and the followers were also scattered; (b) At $t = 20\,\mathrm{s}$, the leaders achieved the prescribed 3D cubic formation and the followers converged inside the convex hull spanned by the leaders.

transient response and a faster convergence rate.

V. Experimental Validation

To demonstrate the effectiveness of the proposed control algorithms, we implemented them on a fleet of Crazyflie 2.1 nano quadcopters and conducted two real-time flight experiments. This section first introduces the Crazyflie 2.1 nano quadcopter and the experimental setup. We then describe the scenarios of the two real-time flight experiments. Finally, we present and discuss the experimental results. A video clip of the real-time flight experiments can be found in the Supplementary Multimedia and at https://youtu.be/Fv1j05rhDoI.

A. Experimental setup

Fig. 6 shows the Crazyflie 2.1 nano quadcopters equipped with expansion decks. The Crazyflie is a compact and lightweight quadcopter, with a diagonal length of $92\,\mathrm{mm}$ from motor to motor and a weight of approximately $27\,\mathrm{g}$, developed as an open-source flying platform by Bitcraze [35]. Additional decks are required to enable indoor autonomous flight experiments, including the Flow Deck v2 and the Loco Positioning system (LPS) deck. The Crazyflie nano quadcopter can estimate its position, velocity and acceleration through the onboard extended Kalman filter and has an onboard inner loop attitude PID controller. In the experiments, the multi-ranger deck is used as an obstacle-detection sensor, enabling the Crazyflie nano quadcopter to detect the distance to obstacles in real time. The LPS, also developed by Bitcraze, provides the absolute position of each Crazyflie nano quadcopter [35].

Remark 5: Since the attitude dynamics of a quadcopter UAV are much faster than its translational dynamics, hovering and manoeuvring can be controlled together by a two-loop control scheme, as shown in Fig. 2 (see [11], [12], [19], [23] and references therein). The distributed controllers in the outer loop drive the networked quadcopters to the prescribed formation/position, while the inner loop control stabilises the attitude dynamics. As a result, the translational dynamics of the Crazyflie nano quadcopter after closing the loop with the inner loop attitude PID controller can be approximated by (1).

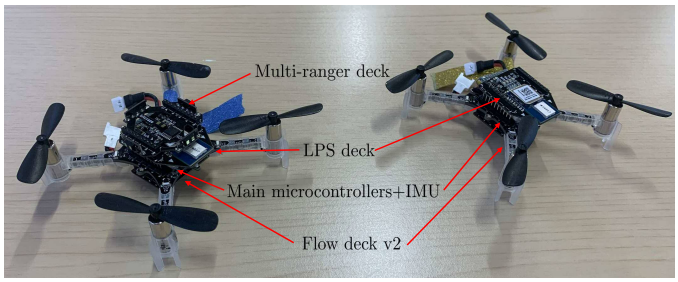
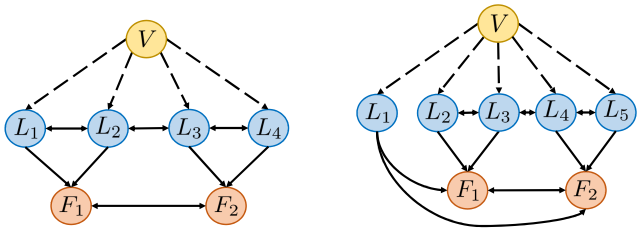


Fig. 6: Crazyflie 2.1 nano quadcopters. We define the Crazyflie nano quadcopter on the left as a leader quadcopter since it has a multi-ranger deck. In contrast, the right one is a follower quadcopter.

We implemented the proposed distributed controllers on a PC station in the real-time flight experiments. The PC station serves as a central hub for collecting state information, including position, velocity, and acceleration, from all Crazyflies and transmitting control commands to them through Crazyradio dongles [35]. However, the PC station can be removed in real-world applications if each quadcopter UAV can directly share its state information with in-neighbours via Bluetooth or Wi-Fi. This is because the proposed controllers in Theorems 1 and 2 are completely distributed.



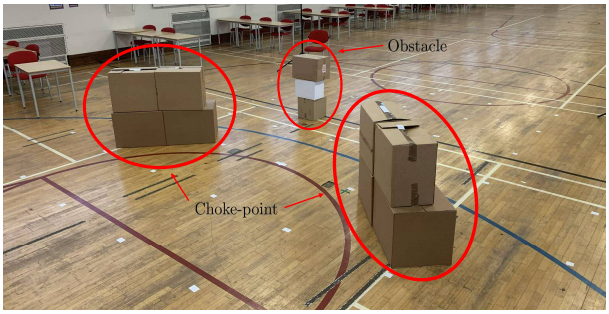
(a) Interaction topology for Experiment 1. (b) Interaction topology for Experiment 2.

Fig. 7: (a) Agents F_1 and F_2 are the followers, agents L_1 to L_4 are the leaders and agent V is the virtual leader agent; (b) Agents F_1 and F_2 are the followers, agents L_1 to L_5 are the leaders and agent V is the virtual leader agent.

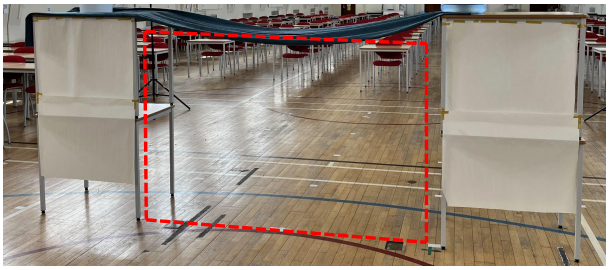
B. Experiment 1: UAVs navigating in an unknown environment

In Experiment 1, six Crazyflie quadcopters, connected via the interaction topology as shown in Fig. 7(a), are used to perform an autonomous flight mission in an unknown environment with a choke point and an obstacle, as shown in Fig. 8(a). However, the multi-ranger decks (i.e. the obstacle-detection sensors) are only attached to the leader quadcopters, while the follower quadcopters do not have any obstacle detection capabilities. The formation configuration is specified so all leader quadcopters achieve a prescribed diamond-shaped formation.

Figs. 9 and 10 present the trajectories of all quadcopters on the X-Y plane and the snapshots of the real-time flight experiment. Initially, the leader agents did not attain the



(a) Flight arena for Experiment 1.



(b) Flight arena for Experiment 2.

Fig. 8: (a) The flight arena with a choke point and an obstacle; (b) A 3D cave-like choke point. The red dashed lines mark the gap of the choke point.

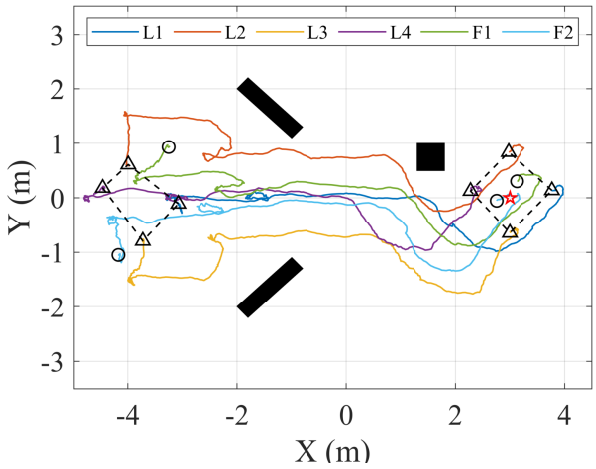


Fig. 9: [Results for Experiment 1:] Trajectories of all quadcopters on the X-Y plane. The red star marks the goal position of the virtual leader agent, while the two black rectangles on the left and the black square on the right represent the choke point and the obstacle. In addition, the circles and triangles mark the positions of the followers and leaders, respectively. The black dashed lines represent the formations of the leaders.

prescribed diamond-shaped formation, and hence, the followers remained unguarded. After that, they achieved and maintained formation-containment control while navigating through the choke point and the obstacle under the action of the proposed control algorithms. Fig. 11 shows the formation tracking error of each leader quadcopter, the containment error of each follower quadcopter and the velocity of each quadcopter in the X-direction. The formation tracking and containment errors converged and remained close to zero after 5 seconds with small oscillations in the steady state, as shown in Figs. 11(a) and (b). It is noted that both formation tracking and containment errors did not converge to exactly zero due to the positioning accuracy of LPS being about $0.1\,\mathrm{m}$. The experimental results validate that networked quadcopters successfully navigated through the choke point and the obstacle and ultimately reached the goal position via the proposed control algorithms. Since all quadcopters consistently maintained formation-containment and tracking behaviours, the follower quadcopters, without any obstacle detection capabilities, also successfully navigated through the choke point and the obstacle.

C. Experiment 2: 3D choke point navigation

In Experiment 2, seven Crazyflie quadcopters, connected via the interaction topology as shown in Fig. 7(b), are used to perform a 3D choke point navigation mission in the flight arena shown in Fig. 8(b). However, only the leader quadcopters are equipped with multi-ranger decks (i.e. the obstacle-detection sensors). In addition, the formation configuration is specified such that all leader quadcopters achieve a prescribed 3D pyramid formation.

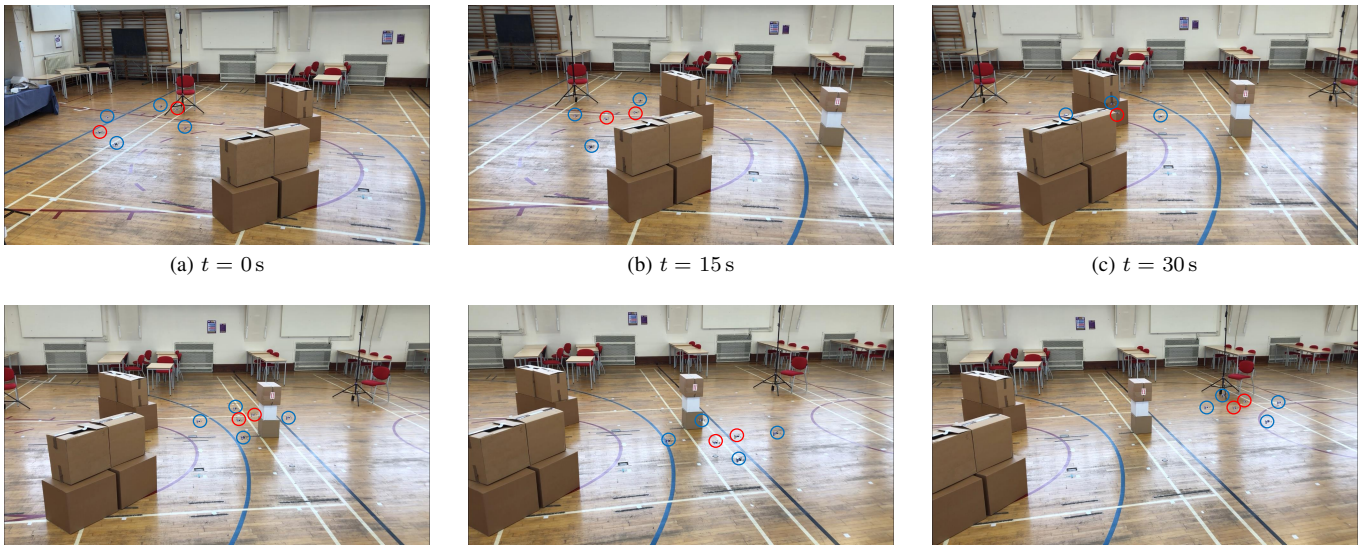
Fig. 12 presents the snapshots of the real-time flight experiment. The initial size of the 3D pyramid formation was bigger than the gap of the choke point, as shown in Fig. 12(c). Under the action of the proposed control algorithms, a team of seven quadcopters maintained formation-containment control while reducing the formation size to navigate through the 3D choke point, as reported in Figs. 12(d) and (e). The experimental results validate that all networked quadcopters successfully travelled in a 3D pyramid formation. Using the proposed control algorithms, they effectively reduced their formation configuration to navigate through the 3D choke point.

VI. Conclusions

This paper has developed a novel hierarchical control framework that addresses the fixed-time formation-containment tracking control problem for MASs while considering motion planning objectives. The proposed control scheme enables networked agents to achieve formation tracking and containment control objectives within a given fixed time. In addition, an effective motion planning strategy is also designed to tackle group obstacle avoidance and choke point navigation challenges, particularly in scenarios when agents do not have prior environmental knowledge and only leader agents have obstacle detection capabilities. To validate the effectiveness and feasibility of the proposed control algorithms, we present a comprehensive MATLAB simulation case study and two real-time flight experiments conducted using a fleet of Crazyflie nano quadcopters.

References

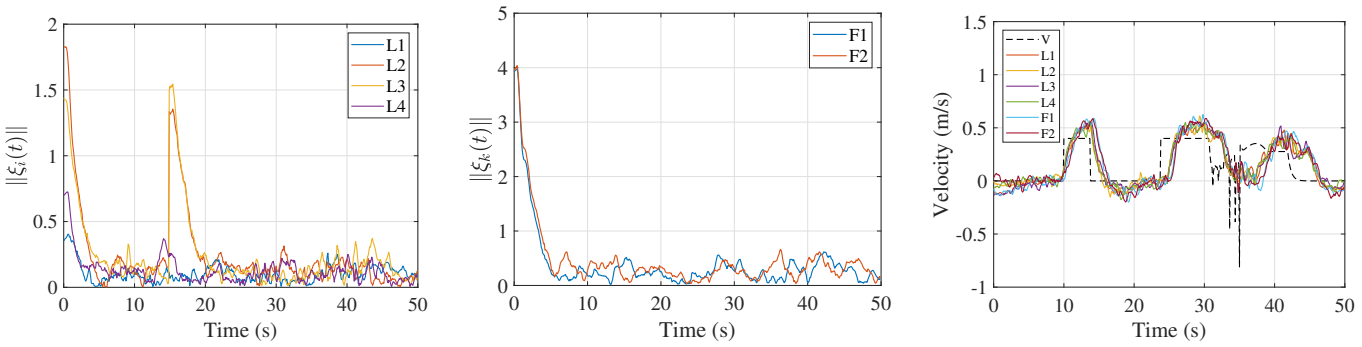
[1] Z. Li and Z. Duan, *Cooperative Control of Multi-Agent Systems : A Consensus Region Approach*, ser. Automation and Control Engineering. Bosa Roca: CRC Press, 2014.



(a) $t = 0\,\mathrm{s}$ (b) $t = 15\,\mathrm{s}$ (c) $t = 30\,\mathrm{s}$

(d) $t = 35\,\mathrm{s}$ (e) $t = 40\,\mathrm{s}$ (f) $t = 50\,\mathrm{s}$

Fig. 10: [Results for Experiment 1:] The snapshots of the real-time flight experiment at $t = 0\,\mathrm{s}$, $t = 15\,\mathrm{s}$, $t = 30\,\mathrm{s}$, $t = 35\,\mathrm{s}$, $t = 40\,\mathrm{s}$ and $t = 50\,\mathrm{s}$. The red circles mark the follower quadcopters, whereas the blue circles mark the leader quadcopters. (a) At $t = 0\,\mathrm{s}$, the leaders started from random initial positions and the followers were also scattered; (b) At $t = 15\,\mathrm{s}$, the leaders achieved the prescribed diamond-shaped formation, and the followers converged inside the convex hull. At the same time, the leaders detected the choke point and updated the scaling variable in the Y-direction; (c) At $t = 30\,\mathrm{s}$, the leaders reduced the formation size such that the whole team could pass through the choke point; (d) At $t = 35\,\mathrm{s}$, the leaders detected the obstacle and all quadcopters started avoiding the obstacle while keeping formation-containment and tracking behaviours; (e) At $t = 40\,\mathrm{s}$, all quadcopters avoided the obstacle and kept moving towards the goal position; (f) At $t = 50\,\mathrm{s}$, all quadcopters reached the goal position, and the mission is complete.



(a) Formation tracking error (b) Containment error (c) Velocity in the X-direction

Fig. 11: [Results for Experiment 1:] (a) Evolution of the formation tracking error $\|\boldsymbol{\xi}_i(t)\|$ of each leader quadcopter. Note that at $t = 15\,\mathrm{s}$, the leader quadcopters detected the choke point, triggering an update to the formation configuration. This led to a reduction in the formation size by the leader quadcopters, causing a sudden increase in the formation tracking error; (b) Evolution of the containment error $\|\boldsymbol{\xi}_k(t)\|$ of each follower quadcopter; (c) Evolution of the velocity of each quadcopter in the X-direction.

[2] S.-J. Chung, A. A. Paranjape, P. Dames, S. Shen, and V. Kumar, "A survey on aerial swarm robotics," *IEEE Transactions on Robotics*, vol. 34, no. 4, pp. 837–855, Aug. 2018.

[3] K. N. McGuire, C. De Wagter, K. Tuyls, H. J. Kappen, and G. C. H. E. de Croon, "Minimal navigation solution for a swarm of tiny flying robots to explore an unknown environment," *Science Robotics*, vol. 4, no. 35, Oct. 2019.

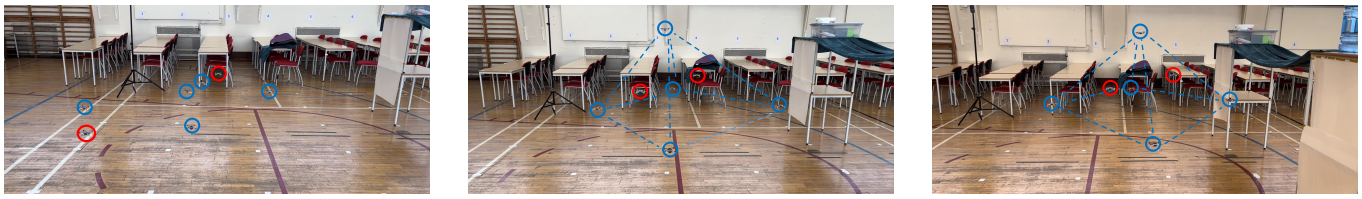
[4] J. Hu, P. Bhowmick, I. Jang, F. Arvin, and A. Lanzon, "A decentralized cluster formation containment framework for multirobot systems," *IEEE Transactions on Robotics*, vol. 37, no. 6, pp. 1936–1955, Dec. 2021.

[5] K. Mohammadi, S. Sirouspour, and A. Grivani, "Control of multiple quad-copters with a cable-suspended payload subject to disturbances," *IEEE/ASME Transactions on Mechatronics*, vol. 25, no. 4, pp. 1709–1718, Aug. 2020.

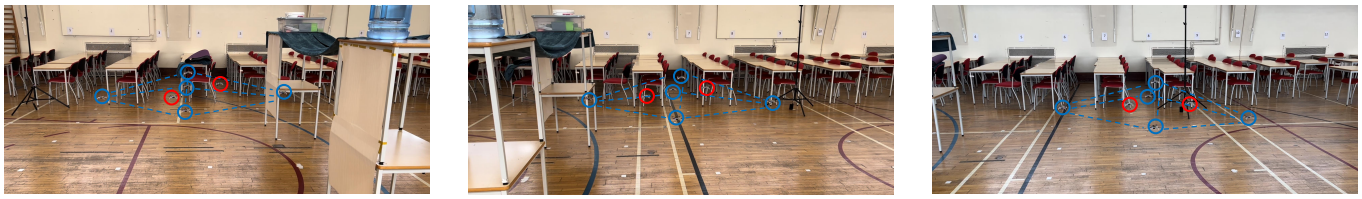
[6] Y.-H. Su, P. Bhowmick, and A. Lanzon, "A robust adaptive formation control methodology for networked multi-UAV systems with applications to cooperative payload transportation," *Control Engineering Practice*, vol. 138, p. 105608, Sep. 2023.

[7] W. Ren, R. W. Beard, and E. M. Atkins, "Information consensus in multivehicle cooperative control," *IEEE Control Systems Magazine*, vol. 27, no. 2, pp. 71–82, Apr. 2007.

[8] Y. Cao, D. Stuart, W. Ren, and Z. Meng, "Distributed containment con-



(a) $t = 0\,\mathrm{s}$ (b) $t = 5\,\mathrm{s}$ (c) $t = 9\,\mathrm{s}$



(d) $t = 14\,\mathrm{s}$ (e) $t = 25\,\mathrm{s}$ (f) $t = 35\,\mathrm{s}$

Fig. 12: [Results for Experiment 2:] The snapshots of the real-time flight experiment at $t = 0\,\mathrm{s}$, $t = 5\,\mathrm{s}$, $t = 9\,\mathrm{s}$, $t = 14\,\mathrm{s}$, $t = 25\,\mathrm{s}$ and $t = 35\,\mathrm{s}$. The red circles mark the follower quadcopters, whereas the blue circles mark the leader quadcopters. The blue dashed lines represent the 3D pyramid formations of leader quadcopters. (a) At $t = 0\,\mathrm{s}$, the leaders started from random initial positions and the followers were also scattered around; (b) At $t = 5\,\mathrm{s}$, the leaders achieved the prescribed 3D pyramid formation, and the followers converged inside the convex hull spanned by the leaders; (c) At $t = 9\,\mathrm{s}$, the leaders detected the choke point and updated the scaling variables in Y and Z directions; (d) At $t = 14\,\mathrm{s}$, the leaders reduced the formation size such that the whole team could pass through the choke point; (e) At $t = 25\,\mathrm{s}$, all quadcopters passed through the choke point fulfilling the formation-containment and tracking objectives; (f) At $t = 35\,\mathrm{s}$, all quadcopters reached the goal position, and the mission is complete.

trol for multiple autonomous vehicles with double-integrator dynamics: algorithms and experiments," *IEEE Transactions on Control Systems Technology*, vol. 19, no. 4, pp. 929–938, 2011.

[9] J. A. Fax and R. M. Murray, "Information flow and cooperative control of vehicle formations," *IEEE Transactions on Automatic Control*, vol. 49, no. 9, pp. 1465–1476, 2004.

[10] J. Wang, A. Lanzon, and I. R. Petersen, "Robust cooperative control of multiple heterogeneous Negative-Imaginary systems," *Automatica*, vol. 61, pp. 64–72, Nov. 2015.

[11] X. Dong, B. Yu, Z. Shi, and Y. Zhong, "Time-varying formation control for unmanned aerial vehicles: theories and applications," *IEEE Transactions on Control Systems Technology*, vol. 23, no. 1, pp. 340–348, 2015.

[12] X. Dong, Y. Zhou, Z. Ren, and Y. Zhong, "Time-varying formation tracking for second-order multi-agent systems subjected to switching topologies with application to quadrotor formation flying," *IEEE Transactions on Industrial Electronics*, vol. 64, no. 6, pp. 5014–5024, 2017.

[13] O. Skeik and A. Lanzon, "Robust output consensus of homogeneous multi-agent systems with negative imaginary dynamics," *Automatica*, vol. 113, p. 108799, Mar. 2020.

[14] H. Liu, T. Ma, F. L. Lewis, and K. P. Valavanis, "Robust time-varying formation control for a set of quad-copters with switching interaction communication topology," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 7, pp. 6880–6890, Jul. 2020.

[15] Y.-H. Su, P. Bhowmick, and A. Lanzon, "A Negative Imaginary Theory-Based Time-Varying Group Formation Tracking Scheme for Multi-Robot Systems: Applications to Quadcopters," in *Proceedings of the IEEE International Conference on Robotics and Automation*, May 2023, pp. 1435–1441.

[16] ——, "Cooperative Control of Multi-Agent Negative Imaginary Systems with Applications to UAVs, Including Hardware Implementation Results," in *Proceedings of the European Control Conference*, Jun. 2023.

[17] ——, "Properties of interconnected negative imaginary systems and extension to formation-containment control of networked multi-UAV systems with experimental validation results," *Asian Journal of Control*, Dec. 2023.

[18] C. Wang, H. Tnunay, Z. Zuo, B. Lennox, and Z. Ding, "Fixed-time formation control of multirobot systems: design and experiments," *IEEE Transactions on Industrial Electronics*, vol. 66, no. 8, pp. 6292–6301, Aug. 2019.

[19] X. Dong, Y. Hua, Y. Zhou, Z. Ren, and Y. Zhong, "Theory and experiment on formation-containment control of multiple multirotor unmanned aerial vehicle systems," *IEEE Transactions on Automation Science and Engineering*, vol. 16, no. 1, pp. 229–240, Jan. 2019.

[20] J. Hu, P. Bhowmick, and A. Lanzon, "Two-layer distributed formation-containment control strategy for linear swarm systems: Algorithm and experiments," *International Journal of Robust and Nonlinear Control*, vol. 30, no. 16, pp. 6433–6453, Nov. 2020.

[21] N. Wang, J. Dai, and J. Ying, "UAV formation obstacle avoidance control algorithm based on improved artificial potential field and consensus," *International Journal of Aeronautical and Space Sciences*, Aug. 2021.

[22] W. Honig, J. A. Preiss, T. K. Kumar, G. S. Sukhatme, and N. Ayanian, "Trajectory planning for quadrotor swarms," *IEEE Transactions on Robotics*, vol. 34, no. 4, pp. 856–869, 2018.

[23] Y.-H. Su and A. Lanzon, "Formation-containment tracking and scaling for multiple quadcopters with an application to choke-point navigation," in *Proceedings of the IEEE International Conference on Robotics and Automation*, May 2022, pp. 4908–4914.

[24] Z. Meng, W. Ren, and Z. You, "Distributed finite-time attitude containment control for multiple rigid bodies," *Automatica*, vol. 46, no. 12, pp. 2092–2099, Dec. 2010.

[25] M. Ji and M. Egerstedt, "Distributed coordination control of multiagent systems while preserving connectedness," *IEEE Transactions on Robotics*, vol. 23, no. 4, pp. 693–703, 2007.

[26] W. Ren, "Multi-vehicle consensus with a time-varying reference state," *Systems & Control Letters*, vol. 56, no. 7-8, pp. 474–483, Jul. 2007.

[27] S. Zhao, "Affine formation maneuver control of multiagent systems," *IEEE Transactions on Automatic Control*, vol. 63, no. 12, pp. 4140–4155, Dec. 2018.

[28] K. Wu, J. Hu, B. Lennox, and F. Arvin, "Finite-time bearing-only formation tracking of heterogeneous mobile robots with collision avoidance,"

IEEE Transactions on Circuits and Systems II: Express Briefs, vol. 68, no. 10, pp. 3316–3320, 10 2021.
[29] Z. Li, H. Tnunay, S. Zhao, W. Meng, S. Q. Xie, and Z. Ding, "Bearing-only formation control with prespecified convergence time," *IEEE Transactions on Cybernetics*, vol. 52, no. 1, pp. 620–629, Jan. 2022.
[30] D. S. Bernstein, *Matrix Mathematics : Theory, Facts, and Formulas*. Princeton: Princeton, N.J: Princeton University Press, 2009.
[31] J. Barraquand and J.-C. Latombe, "Robot Motion Planning: A Distributed Representation Approach," *The International Journal of Robotics Research*, vol. 10, no. 6, pp. 628–649, Dec. 1991.
[32] M. Mabrouk and C. McInnes, "Solving the potential field local minimum problem using internal agent states," *Robotics and Autonomous Systems*, vol. 56, no. 12, pp. 1050–1060, Dec. 2008.
[33] K. Chang, Y. Xia, and K. Huang, "UAV formation control design with obstacle avoidance in dynamic three-dimensional environment," *SpringerPlus*, vol. 5, no. 1, p. 1124, Dec. 2016.
[34] D. Choi, K. Lee, and D. Kim, "Enhanced Potential Field-Based Collision Avoidance for Unmanned Aerial Vehicles in a Dynamic Environment," in *AIAA Scitech 2020 Forum*, vol. 1 PartF. American Institute of Aeronautics and Astronautics, Jan. 2020.
[35] Bitcraze, "https://www.bitcraze.io."

Yu-Hsiang Su received his B.Sc. degree in Mechanical Engineering from National Chiao Tung University, Hsinchu, Taiwan, in 2017 and his M.Sc. degree in Advanced Control and Systems Engineering from the University of Manchester, Manchester, UK, in 2019. He is currently pursuing his PhD in Electrical and Electronic Engineering at the University of Manchester, UK. His research interests include cooperative control of multi-agent systems and applications in multi-UAV systems.

Parijat Bhowmick is presently an Assistant Professor at the Dept. of EEE of IIT Guwahati, India. Before joining this position, he worked for three years as a Post-doctoral Research Associate at the Control Systems Centre, University of Manchester. He completed his PhD in Control Engineering from IIT Kharagpur, India 2018. He did his Master's from Jadavpur University in 2012 with the same specialisation. He received the University Gold Medal in his Masters and the Institute Silver Medal during his Bachelor's. He is an active researcher in the horizon of Robust control of uncertain systems, Negative Imaginary Systems theory, Vibration control of mechatronic systems, Cooperative Control of Multi-agent Systems (including multi-agent Systems) and Control of Smart/Micro-grid Systems.

Alexander Lanzon received his Ph.D. degree in Control Engineering and his M.Phil. degree in Robot Control from the University of Cambridge, Cambridge, UK, in 2000 and 1997 respectively and received his B.Eng.(Hons). degree in Electrical and Electronic Engineering from the University of Malta, Msida, Malta, in 1995.

He has held research and academic positions at Georgia Institute of Technology, Atlanta GA, USA, and the Australian National University, Canberra ACT, Australia, and industrial positions at ST-Microelectronics Ltd., Kirkop, Malta; Yaskawa Denki (Tokyo) Ltd., Saitama-Ken, Japan; and National ICT Australia Ltd., Canberra ACT, Australia. In 2006, he joined the University of Manchester, Manchester, UK, where he now holds the Chair in Control Engineering. His research interests include the fundamentals of feedback control theory, negative imaginary systems theory, robust control theory, input-output nonlinear control theory, and applying robust and nonlinear control theory to innovative mechatronic, robotic and drone applications.

Prof. Lanzon is a Fellow of the Institute of Mathematics and its Applications, the Institute of Measurement and Control and the Institution of Engineering and Technology. He is currently an Associate Editor of the IEEE Control Systems Letters (L-CSS). He has served as an Associate Editor of the IEEE Transactions on Automatic Control from 2012 to 2018, and as a Subject Editor of the International Journal of Robust and Nonlinear Control from 2012 to 2015.