Self-Reconfigurable Hierarchical Frameworks for Formation Control of Robot Swarms

Yuwei Zhang, Sinan Oğuz, Shaoping Wang, Emanuele Garone, *Member, IEEE*, Xingjian Wang, Marco Dorigo, *Fellow, IEEE*, and Mary Katherine Heinrich

***Abstract*—Hierarchical frameworks—a special class of directed frameworks with a layer-by-layer architecture—can be an effective mechanism to coordinate robot swarms. Their effectiveness was recently demonstrated by the mergeable nervous systems paradigm (Mathews et al., 2017), in which a robot swarm can switch dynamically between distributed and centralized control depending on the task, using self-organized hierarchical frameworks. New theoretical foundations are required to use this paradigm for formation control of large swarms. In particular, the systematic and mathematically analyzable organization and reorganization of hierarchical frameworks in a robot swarm is still an open problem. Although methods for framework construction and formation maintenance via rigidity theory exist in the literature, they do not address cases of hierarchy in a robot swarm. In this article, we extend bearing rigidity to directed topologies and extend the Henneberg constructions to generate self-organized hierarchical frameworks with bearing rigidity. We investigate three-key self-reconfiguration problems: 1) framework merging; 2) robot departure; and 3) framework splitting. We also derive the mathematical conditions of these problems and then develop algorithms that preserve rigidity and hierarchy using only local information. Our approach can be used for formation control generally, as in principle it can be coupled with any control law that makes use of bearing rigidity. To demonstrate and validate our proposed hierarchical frameworks and methods, we apply them to four scenarios of reactive formation control using an example control law.**

***Index Terms*—Aerial swarm, AUVs, bearing rigidity, formation control, hierarchical framework, mobile robots, rigidity maintenance, robot swarm, UAVs, underwater swarm.**

Manuscript received 10 February 2022; revised 4 August 2022 and 8 December 2022; accepted 3 January 2023. Date of publication 3 February 2023; date of current version 19 December 2023. The work of Marco Dorigo and Mary Katherine Heinrich was supported by Belgian F.R.S.-FNRS. This work was supported in part by the National Science and Technology Major Project under Grant 2017-V-0010-0060; in part by the National Natural Science Foundation of China under Grant 51620105010 and Grant 51675019; in part by the National Basic Research Program of China under Grant JCKY2018601C107; in part by the Fellowship of China Postdoctoral Science Foundation under Grant 2022M710305; in part by the Program of Concerted Research Actions (ARC) of the Université Libre de Bruxelles; and in part by the Office of Naval Research Global under Award N62909-19-1-2024. This article was recommended by Associate Editor P. Shi. *(Corresponding author: Shaoping Wang.)*

Yuwei Zhang is with the School of Automation Science and Electrical Engineering, Beihang University, Beijing 100191, China (e-mail: zhangyuwei@buaa.edu.cn).

Sinan Oğuz is with the Institut de Recherches Interdisciplinaires et de Développements en Intelligence Artificielle, Université Libre de Bruxelles, 1050 Brussels, Belgium, and also with the Unité d'enseignement en Automatique et Analyse des Systémes, Université Libre de Bruxelles, 1050 Brussels, Belgium.

Shaoping Wang and Xingjian Wang are with the School of Automation Science and Electrical Engineering, Beihang University, Beijing 100191, China (e-mail: shaopingwang@buaa.edu.cn).

Emanuele Garone is with the Unité d'enseignement en Automatique et Analyse des Systémes, Université Libre de Bruxelles, 1050 Brussels, Belgium.

Marco Dorigo and Mary Katherine Heinrich are with the Institut de Recherches Interdisciplinaires et de Développements en Intelligence Artificielle, Université Libre de Bruxelles, 1050 Brussels, Belgium.

Color versions of one or more figures in this article are available at https://doi.org/10.1109/TCYB.2023.3237731.

Digital Object Identifier 10.1109/TCYB.2023.3237731

I. Introduction

In THE last decades, multirobot systems have been proposed as the default solution to carry out certain classes of missions [1], such as cooperative object transportation [2] and search and rescue [3]. For any mission, the robots' performance depends on the suitability of the chosen control strategy for the given task. It is well known that centralized control of large multirobot systems poses several problems, including limited scalability, a single point of failure in the coordinating agent, and potentially unrealistic communication infrastructures. To circumvent these problems, the swarm robotics community has successfully demonstrated that groups of robots can be controlled in a completely decentralized way [4], [5], [6], [7], [8], [9]. However, as the size and speed of a fully decentralized swarm increases, the design and management of swarm-level behaviors become increasingly difficult. To address this dichotomy in the context of formation control, we study formation frameworks that are self-organized and use only local measurement and communication, but still, incorporate some aspects of centralization to improve manageability. We propose a systematic way to build these self-reconfigurable hierarchical frameworks and use them to support formation control laws.

A. Motivations

Formation control in challenging environments—such as underwater, underground, or inside buildings with unknown interiors—is not yet fully understood [10]. In such environments, formation control cannot rely on absolute references (e.g., a global reference frame), because external positioning infrastructures, such as GPS or off-board sensing, are often not available. According to the recent survey by [11], current approaches that use only relative information generally have the following limitations: they either 1) require a specific topology or connection property but have not thoroughly addressed connectivity preservation or 2) use fully decentralized control, which complicates global and local stability,

but have not thoroughly addressed stability properties. We aim to address these two gaps by designing self-organized frameworks with mathematically provable properties to support formation controllers. The frameworks we propose use self-reconfigurability to preserve connectivity and use self-organized hierarchy to avoid the stability issues of full decentralization.

We aim to design frameworks that can be comprehensively self-reconfigured on demand, not only for expected mission requirements but also for unexpected disturbances. Environmental conditions, for instance, might suddenly require formations to grow or shrink in size. As an example, a formation might need to split into multiple subformations to fit through a narrow passage, then merge to resume the original mission. Formations might also need to reconfigure in order to recover from robot malfunctions. For example, if several robots fail suddenly due to a collision, the formation will need to reconfigure so as to maintain connectivity and stability after those departures. In this article, we, therefore, address the component operations of all framework construction and reconstruction problems: robot addition, framework merging, robot departure, and framework splitting.

B. Related Work

The idea of self-reconfigurable hierarchical frameworks has recently been introduced in the literature as mergeable nervous systems (MNSs) [1], [12], [13]. The main idea of the MNS approach is to control a swarm through a self-organized hierarchical control framework, where both the "brain" robot and the communication hierarchy are determined dynamically and are self-reconfigurable. The MNS approach allows a robot swarm to adjust the degree of decentralization used in its control strategy, based on the appropriateness for a given task. So far, the practical effectiveness of the MNS approach has been demonstrated for small groups of robots [1], [12], [13]. In order for the MNS paradigm to extend to formation control of much larger robot swarms and swarms that include fast ground vehicles or drones, new theoretical foundations need to be developed to complement the existing practical studies. There have been a few studies in which robots join a hierarchy using local decisions (e.g., [9]), but construction of self-organized hierarchical frameworks is currently not fully understood.

Rigidity graph theory is a fundamental mathematical tool to handle various problems in networked robotic systems, e.g., [14] and [15]. Our approach to hierarchical frameworks is based on the concept of bearing rigidity, which has recently been used to address network problems in formation control [16], [17], [18], [19], [20], [21], [22], [23] and network localization [24]. *Bearing rigidity* is a graph property that allows a formation to be maintained without external positioning, using only interagent measurements of bearing. Bearing (i.e., vector of arrival) can be sensed directly by onboard cameras or sensor arrays [18] in conjunction with an onboard inertial measurement unit. Bearing vectors remain unchanged during translational and scaling maneuvering of the formation [16], enabling high flexibility information management. However, the underlying graphs in existing approaches are assumed to be undirected. These are less natural than directed graphs when dealing with multirobot systems. If the graph is assumed to be undirected, then constant mutual visibility among all robot pairs needs to be ensured. In practice, this cannot be guaranteed, as some communication breaks will be unavoidable, regardless of the sensing type. Although bearing rigidity under undirected graphs has been developed in [25], it is heavily based on symmetry, and, therefore, cannot be simply applied to directed graphs [26]. Bearing-only formation maneuvering under directed graphs with hierarchical structures was considered in [27], but rigidity was not analyzed.

Within bearing rigidity, *infinitesimal bearing rigidity* is the most important notion. In general terms, infinitesimal bearing rigidity implies that each robot can find its unique target position using only inter-robot measurements of bearing vectors. A predominant algorithm for constructing sequentially infinitesimally bearing rigid (IBR) frameworks is the so-called Henneberg construction. Originally proposed for distance rigidity [28], the Henneberg constructions have been extended to bearing rigidity in [26] and [29]. The sequential nature of a Henneberg construction lends itself well to the determination of hierarchy, including for the design of hierarchical control structures for swarms. Based on this idea, in this article, we propose a novel approach to swarm control that combines the use of self-organized hierarchy (see MNS [1]) with the Henneberg construction. Our approach has provable reconfiguration properties and allows for formation control in swarms using behaviors that are simple to design.

Besides the self-reconfiguration of frameworks, which is a crucial feature of this approach, it is also important to investigate the preservation of rigidity in these scenarios. Attempts to solve the rigidity recovery problem can be found in the literature, in scenarios of merging [30], robot departure [31], [32], and splitting [33]. However, in order to add or remove the minimum edges to maintain rigidity, these existing solutions use global assessment and require centralized control. Note that approaches that depend on global assessment suffer from poor scalability, and, therefore, cannot be applied to large-scale robot swarms. This further motivates us to design distributed self-reconfiguration algorithms.

C. Approach and Contributions

This article addresses self-reconfigurable hierarchical frameworks in robot swarms, for the purpose of formation control, enabled by bearing rigidity theory and the notion of hierarchy. Our hierarchical frameworks can construct and reconstruct themselves comprehensively (through robot addition, framework merging, robot departure, and framework splitting) using only local measurement and local communication. To demonstrate our hierarchical frameworks and validate our theoretical results, we apply them with a simple example controller and run experiments in four scenarios.

The main technical contributions of this article can be summarized as follows.

1) We address a current gap in bearing-based approaches by extending to multirobot systems that communicate over

directed networks. Different from existing approaches that are restricted to undirected networks [16], [17], [18], [19], [20], [21], [24], we extend bearing rigidity theory to directed graphs with lower-triangular structure. This is nontrivial because of the analysis difficulty associated with the lack of symmetry, which must be handled in directed graphs. We show that to evaluate rigidity under such a topology, in addition to the concept of *infinitesimal bearing rigidity* given in [25], the concept of *bearing persistence* is required. We provide the necessary and sufficient conditions to uniquely determine a framework under a directed topology with asymmetric and lower-triangular structure.

2) We propose a novel hierarchical Henneberg construction (HHC) to integrate the concepts of bearing rigidity and hierarchy. Compared to the bearing-based Henneberg Construction proposed in [26], which analyzes rigidity from a geometric perspective using a global reference frame, our method can establish rigidity intrinsically and in a decentralized way (i.e., without global references). The only existing approach to analyze bearing rigidity in a decentralized way without a global reference is [10], which has few constraints on topology, but still, assumes that the topology has already been established. Our approach, by contrast, uses hierarchy to construct and maintain the necessary topology and intrinsically establish rigidity, without relying on an absolute reference, system-wide broadcast, or other global mechanisms.
3) We propose the mathematical conditions and design the distributed algorithms for framework construction (including robot addition), framework merging, robot departure, and framework splitting. Compared with [30], [31], [32], and [33], which rely on global assessment to determine the reconfiguration strategies, our methods can be implemented in a distributed manner due to the hierarchical structure. It is worth noting that our framework reconfiguration is not a simple topological sorting because both the bearing rigidity and hierarchy of the framework should be preserved.

The remainder of this article is organized as follows. In Section II, the foundational concepts of bearing rigidity and bearing persistence are presented. In Section III, we formulate three-key problems addressed in this article: 1) framework construction; 2) framework reconstruction; and 3) validation of our frameworks using an example formation control law. These problems are addressed in Sections IV–VI, respectively. The conclusions are summarized in Section VII.

II. From Bearing Rigidity to Bearing Persistence

Notation: $\mathbb{R}^d$ is the d-dimensional Euclidean space. $\mathbf{0}$ is a zero matrix with appropriate dimension; $d \times d$ identity matrix is denoted by $\mathbf{I}_d$ and $n \times 1$ vector of all ones is denoted by $\mathbf{1}_n$. rank$(\cdot)$ and Null$(\cdot)$ are the rank and null space of a square matrix; card$(\cdot)$ denotes the number of elements in a set. $\|\cdot\|$ is the Euclidean norm of a vector.

Consider a set of n ($n \geq 2$) robots in $\mathbb{R}^d$ ($d = 2, 3$). $p_i(t) \in \mathbb{R}^d$ denotes the position of robot $i \in \{1, 2, \ldots, n\}$ at time t and the vector $p(t) = [p_1^T(t), p_2^T(t), \ldots, p_n^T(t)]^T \in \mathbb{R}^{dn}$ describes the configuration of the robot swarm at time t. Interactions among the robots are characterized by a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where $|\mathcal{V}| = n$ and $|\mathcal{E}| = m$. If $e_{ji} = (e_j, e_i) \in \mathcal{E}$, then the ith robot can receive information from the jth robot. $\mathcal{G}$ is undirected if $\forall e_{ji} \in \mathcal{E}$, there exists $e_{ij} \in \mathcal{E}$; otherwise, $\mathcal{G}$ is directed. We define the parent set of vertex v_i as $\mathcal{P}_i = \{v_j \in \mathcal{V} | e_{ji} \in \mathcal{E}\}$, and the child set of v_i as $\mathcal{C}_i = \{v_j \in \mathcal{V} | e_{ij} \in \mathcal{E}\}$.

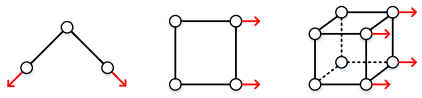

Fig. 1. Examples of *non-IBR* frameworks. The red arrows represent nontrivial infinitesimal bearing motions, under which the framework will deform and cannot be uniquely determined.

It is assumed that for each edge $e_{ji} \in \mathcal{E}$, robot i can continuously measure the bearing of robot j where the bearing vector is $g_{ij} = p_{ij}/\|p_{ij}\|$ and where $p_{ij} = p_j - p_i$ is the displacement vector.

We define a *framework* as a graph $\mathcal{G}$ associated with a configuration p, i.e., $(\mathcal{G}, p)$. According to whether the underlying graph is directed or not, the framework is either an *undirected framework* or a *directed framework*.

In the next section, we will recall some classical concepts concerning the so-called bearing rigidity of undirected frameworks, and in the subsequent section, we will report a series of new results on directed frameworks that will be used in this article.

A. Bearing Rigidity in Undirected Frameworks

To describe all the bearings in $(\mathcal{G}, p)$, define the bearing function F_B as $F_B(p) \triangleq [g_1^T, g_2^T, \ldots, g_m^T]^T$, where g_k corresponds to the kth edge in graph $\mathcal{G}$. Then, we can define the *bearing rigidity matrix* as

$$R_B(p) \triangleq \frac{\partial F_B}{\partial p} \in \mathbb{R}^{dm \times dn}. \tag{1}$$

Definition 1 (Infinitesimal Bearing Rigidity [25]): An undirected framework $(\mathcal{G}, p)$ in $\mathbb{R}^d$ is IBR if and only if the positions of all robots in the framework can be uniquely determined up to a translational and scaling factor.

Lemma 1 [25]: An undirected framework $(\mathcal{G}, p)$ in $\mathbb{R}^d$ is IBR if and only if $\text{rank}(R_B(p)) = dn - d - 1$, or equivalently $\text{Null}(R_B(p)) = \text{span}\{\mathbf{1}_n \otimes \mathbf{I}_d, p\}$.

Another equivalent definition for an IBR framework is that all the infinitesimal bearing motions are trivial,[1] i.e., translation and scaling are the only robot motions that preserve the relative bearings between robots connected by an edge. Examples of *noninfinitesimally bearing rigid frameworks* are presented in Fig. 1, where there clearly exist nontrivial infinitesimal bearing motions (see red dashed arrows), under which the framework will deform. By contrast, Fig. 2 shows examples of rigid frameworks where the only infinitesimal motions possible are rigid translation and scaling of

[1]Two kinds of trivial infinitesimal bearing motions exist: translational and scaling of the entire framework. More details are given in [25].

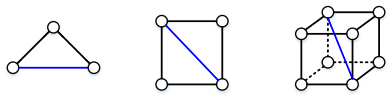

Fig. 2. Examples of *IBR* frameworks. In contrast to examples in Fig. 1, the newly added edges (see blue edges) can eliminate nontrivial infinitesimal bearing motions, such that the configuration is uniquely determined.

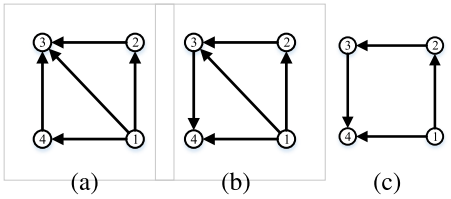

Fig. 3. Examples to illustrate the concepts of bearing rigidity [34] and bearing persistence in directed frameworks. (a) *IBR* but *non-BP* framework. (b) *IBR* and *BP* framework. (c) *Non-IBR* but *BP* framework.

the frameworks. Note that the cases reported in Fig. 2 are obtained by rigidifying the examples in Fig. 1, by adding edges (see blue edges) to eliminate nontrivial infinitesimal bearing motions.

B. Bearing Persistence in Directed Frameworks With Lower-Triangular Structure

It is worth noting that rigidity is fundamentally an *undirected* notion, and, therefore, is not sufficient to characterize *directed* frameworks [35]. Consider the framework in Fig. 3(a). Although it is IBR (because of the rank of R_B), this framework cannot always be determined uniquely. In this framework, robot 1 has no bearing constraints, therefore, it can be placed arbitrarily in space. After the position of robot 1 is determined, robots 2 and 4 can be subsequently placed. However, robots 2 and 4 only have one bearing constraint and they can be randomly placed along edges e_{12} and e_{14}. Once the positions of robots 1, 2, and 4 are determined, it is clear that the position of robot 3 is not always feasible, because it has three bearing constraints to be satisfied. The position of robot 3 is feasible if and only if $\|p_{21}\| = \|p_{41}\|$. Therefore, rigidity is not sufficient to characterize the framework in Fig. 3(a). By contrast, the framework in Fig. 3(b) can be uniquely determined as an undirected framework.

This example indicates that more conditions are required to guarantee the existence and uniqueness of a directed framework. Therefore, in this article, we will also use the condition of *bearing persistence*. Before defining this notion, we introduce another bearing-related matrix $B \in \mathbb{R}^{dn\times dn}$, namely the *bearing Laplacian*, which is defined as [25]

$$B_{ij} = \begin{cases} \mathbf{0}, & i \neq j, e_{ji} \notin \mathcal{E} \\ -P_{g_{ij}}, & i \neq j, e_{ji} \in \mathcal{E} \\ \sum_{v_k \in \mathcal{P}_i} P_{g_{ik}}, & i = j \end{cases} \tag{2}$$

where $B_{ij} \in \mathbb{R}^{d\times d}$ is the ijth block of a submatrix of B, and $P_{g_{ij}}$ is an orthogonal projection operator defined as $P_{g_{ij}} \triangleq \mathbf{I}_d - g_{ij}g_{ij}^T$. It can be proved that $P_{g_{ij}}$ is positive semi-definite, 0 is a simple eigenvalue of $P_{g_{ij}}$, $\text{Null}(P_{g_{ij}}) = \text{span}(p_i - p_j)$, and $\text{rank}(P_{g_{ij}}) = d - 1$.

Lemma 2 [36]: $\text{rank}(B_{ii}) = d$ if and only if there exist at least two vertices $v_j, v_k \in \mathcal{P}_i$ such that $g_{ij} \neq g_{ik}$.

We can now introduce the definition of bearing persistence.

Definition 2 (Bearing Persistence [34]): A directed framework $(\mathcal{G}, p)$ in $\mathbb{R}^d$ is bearing persistent (BP) if $\text{Null}(B) = \text{Null}(R_B)$.

For *undirected* frameworks, the bearing Laplacian matrix B is symmetric positive semi-definite, which satisfies $\text{Null}(B) = \text{Null}(R_B)$ [34]. For *directed* frameworks, however, only $\text{Null}(R_B) \subset \text{Null}(B)$ is guaranteed. Note that bearing persistence is independent of rigidity. An example is illustrated in Fig. 3(c), which is not IBR but is still BP.

Even when using persistence, whether all IBR and BP-directed frameworks can be uniquely determined is still an open problem [25]. This research focuses on directed graphs with a hierarchical structure. Specifically, each robot only observes and tracks two immediate neighbors with higher hierarchy (i.e., parents), which results in a lower-triangular structure (see [37]). Therefore, the bearing Laplacian of these special directed graphs can be written as

$$B = \begin{bmatrix} \mathbf{0} & & & \mathbf{0} \\ B_{2,1} & B_{2,2} & & \\ \vdots & \vdots & \ddots & \\ B_{n,1} & B_{n,2} & \cdots & B_{n,n} \end{bmatrix}. \tag{3}$$

Lemma 3: Consider a directed framework $(\mathcal{G}, p)$ in $\mathbb{R}^d$. If the corresponding bearing Laplacian matrix B is lower triangular, the following statements are equivalent.

1) $(\mathcal{G}, p)$ is IBR and BP.
2) $(\mathcal{G}, p)$ can be uniquely determined up to a translational and scaling factor.
3) $\text{Null}(B) = \text{Null}(R_B) = \text{span}\{\mathbf{1}_n \otimes \mathbf{I}_d, p\}$.
4) $\text{rank}(B) = dn - d - 1$.
5) $\text{rank}(B_{2,2}) = d - 1$ and $\text{rank}(B_{ii}) = d \quad \forall i \geq 3$.

The proof of Lemma 3 is reported in Appendix A. Lemma 3 gives necessary and sufficient conditions to uniquely determine a directed framework with a lower triangular matrix B. Note that the structure of B can be different under distinct labeling rules. Here, we only require that one labeling rule exists, such that B is lower triangular, then Lemma 3 will be applicable immediately.

Infinitesimal bearing rigidity and bearing persistence are generic properties, which are mainly determined by the structure of the underlying graph, rather than the configuration. To highlight this fact, we introduce the following definition.

Definition 3 [Generically Bearing Rigid and BP (GBR-BP) Graph]: A directed graph $\mathcal{G}$ is GBR-BP in $\mathbb{R}^d$ if there exists at least one configuration $p \in \mathbb{R}^{dn}$ such that $(\mathcal{G}, p)$ in $\mathbb{R}^d$ is IBR and BP.

Lemma 4: Consider a directed graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, with a lower triangular bearing Laplacian matrix B. $\mathcal{G}$ is GBR-BP if and only if $\text{card}(\mathcal{P}_2) = 1$ and $\text{card}(\mathcal{P}_i) \geq 2 \quad \forall i \geq 3$.

The proof of Lemma 4 is given in Appendix B. Lemma 4 provides an admissible solution to construct directed GBR-BP graphs and provides the theoretical basis needed to develop construction and reconfiguration strategies later. GBR-BP graphs have the following two properties.

Lemma 5: Consider a GBR-BP graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, with a lower-triangular structure. Add an edge e_{ji} to the graph $\mathcal{G}$, where $v_i, v_j \in \mathcal{V}$ and $j < i$. The resultant graph $\mathcal{G}^+ = (\mathcal{V}, \mathcal{E}^+)$ with $\mathcal{E}^+ = \mathcal{E} \cup \{e_{ji}\}$ is GBR-BP.

Lemma 6: Consider a GBR-BP graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, with a lower-triangular structure, delete an edge $e_{ki} \in \mathcal{E}$, and add an edge $e_{ji} \notin \mathcal{E}$ with $j < i$. Then, the resultant graph $\mathcal{G}' = (\mathcal{V}, \mathcal{E}')$ with $\mathcal{E}' = (\mathcal{E} \backslash e_{ki}) \cup \{e_{ji}\}$ is GBR-BR.

Lemmas 5 and 6 can be directly derived from Lemma 4, and, thus, the proofs are omitted. In other words, Lemma 5 allows us to connect new parent vertices to any vertex v_i $(i \geq 3)$. Lemma 6 allows us to change the parent vertices of any vertex v_i $(i \geq 3)$. This provides us flexibility in adjusting the topology of a robot swarm dynamically, while the bearing rigidity and persistence are guaranteed.

III. Problem Statement

Based on the concepts of bearing rigidity and bearing persistence, the objective of this article is to investigate the construction and reconstruction of self-reconfigurable hierarchical frameworks in a robot swarm. The following three questions will be addressed.

1) Given a swarm of n robots capable of onboard bearing measurements, how can the robots construct a hierarchical and GBR-BP graph?
2) Given the constructed graph, how can the hierarchy and rigidity properties be preserved in self-reconfiguration scenarios, specifically in *merging of frameworks*, *robot departure*, and *splitting of frameworks*?
3) Given the hierarchical frameworks, when coupled with an example control law that makes use of bearing rigidity, how can the robot swarm achieve and reconfigure an arbitrary target formation with moving leaders while preserving the hierarchy and rigidity properties during self-reconfiguration scenarios?

IV. Framework Construction

An important precondition to use Lemma 4 is that the bearing Laplacian of the framework is lower triangular. In this section, we extend the Henneberg constructions by introducing the notion of *hierarchy*, which not only guarantees the rigidity and persistence requirement but also ensures the lower-triangular feature of the bearing Laplacian. Our proposed algorithm is inspired by [26] and is defined as follows.

HHC: Consider a group of n robots $(n \geq 2)$. The first step is to arbitrarily choose two robots in the group as the leader robots, denoted by v_1 and v_2, and add an edge $e_{1,2}$ connecting them. Define the hierarchy $h(v_i)$ of a generic robot v_i as the length of its longest path from v_i to v_1 in the directed graph $\mathcal{G}$. The hierarchy of v_1 and v_2 is 0 and 1, respectively, i.e., $h(v_1) = 0$, $h(v_2) = 1$. In subsequent steps, we utilize one of the following two operations.

1) *Vertex Addition:* Add a new vertex v_i to the existing graph, incorporating two directed edges e_{ji} and e_{ki} to two existing vertices v_j and v_k in the graph. Then, the hierarchy of vertex v_i is defined as $h(v_i) = \max(h(v_j)), h(v_k) + 1$.

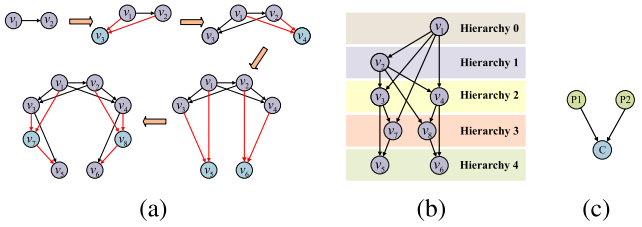

Fig. 4. Example of HHC for a group of eight robots. (a) Four steps of construction, with the added vertex and edges in blue and red, respectively. Vertex addition is employed in steps 1–3 and edge splitting is used in step 4. (b) Hierarchy layers of the framework resulting from the construction process in (a), in which the two leaders are on the first and second layers (i.e., 0 and 1). (c) Minimal structure, where P1 and P2 are parents and C is the child.

2) *Edge Splitting:* Consider an existing vertex v_k in the graph, which has two parents v_j and v_p in the graph. Remove an edge e_{jk} from the graph and add a new vertex v_i together with three edges e_{ik}, e_{ji}, and e_{li}, where vertex v_l is selected such that $h(v_l) \leq h(v_k)$. Then, update the hierarchy of v_i as $h(v_i) = \max(h(v_j), h(v_l)) + 1$ and the hierarchy of v_k as $h(v_k) = \max(h(v_i), h(v_p)) + 1$.

An example of HHC for a group of eight robots is presented in Fig. 4(a). An important feature of HHC is that all the robots, except the two that are arbitrarily selected as leaders, have exactly two parents. Moreover, each follower can form a connection with each of its two parents, forming a *minimal structure*, as shown in Fig. 4(c). The child receives commands from its parents and obtains its parents' states via communication or sensing, and then uses this information to coordinate with its parents. On the basis of the hierarchical framework, shown in Fig. 4(b), the framework can also be viewed as an acyclic tree, with the first of the two leaders as the root.

We define a layer-by-layer labeling rule to verify that the bearing Laplacian matrix of a framework generated by HHC is lower triangular. Let n_l denote the number of vertices with hierarchy l. Vertices with hierarchy 0 are labeled from v_1 to v_{n_0}. Vertices with hierarchy $l \geq 1$ are labeled from $v_{n_{l-1}+1}$ to v_{n_l}. Note that there is no order requirement when labeling vertices with the same hierarchy layer. Based on this labeling rule, the bearing Laplacian B can be rewritten as

$$B = \begin{bmatrix} \mathbf{0} & & & & & \mathbf{0} \\ B_{2,1} & B_{2,2} & & & & \\ \vdots & \vdots & \vdots & \ddots & & \\ \mathbf{0} & -B_{ij} & \mathbf{0} & -B_{ik} & B_{ii} & \\ \vdots & \vdots & \vdots & \vdots & \vdots & \ddots \end{bmatrix}. \tag{4}$$

A graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ generated via Henneberg construction is called a *Laman graph* [38]. It was proved in [36] that an *undirected Laman graph* is generically bearing rigid. Here, we further show that a *directed Laman graph* is GBR-BP.

Theorem 1: A graph $\mathcal{G}$, generated by HHC, is GBR-BP.

Proof: Following Lemma 4, $\mathcal{G}$ is GBR-BP if and only if $\text{card}(\mathcal{P}_2) = 1$ and $\text{card}(\mathcal{P}_i) \geq 2 \quad \forall i \geq 3$. Denote the graph consisting of n vertices as $\mathcal{G}_n = (\mathcal{V}_n, \mathcal{E}_n)$.

First, we consider the case of $n = 2$. $\mathcal{G}_2 = (\mathcal{V}_2, \mathcal{E}_2)$ is defined as $\mathcal{V}_2 = \{v_1, v_2\}$ and $\mathcal{E}_2 = \{e_{1,2}\}$. Note that the bearing

Laplacian matrix of $\mathcal{G}_2$ is lower triangular, and $\text{card}(\mathcal{P}_2) = 1$. Therefore, the claim is true for $n = 2$.

Second, suppose that the claim is true for $2 \leq l \leq n-1$. Now, we consider the case of $l = n$, i.e., a new vertex v_n will be added to $\mathcal{G}_{n-1}$. According to HHC, there are the following two cases.

Vertex Addition: Select two distinct vertices v_j and v_k from $\mathcal{G}_{n-1}$. We add edges e_{jn} and e_{kn}. It is trivial to verify that the bearing Laplacian matrix is still lower-triangular, as $j, k < n$. Moreover, given that $\mathcal{G}_{n-1}$ is GBR-BP and $\text{card}(\mathcal{P}_n) = 2$, then $\mathcal{G}_n$ is GBR-BP under Lemma 4.

Edge Splitting: We select three vertices v_k, v_j, and v_l from $\mathcal{G}_{n-1}$, according to the requirements specified in the operation description. Then, the new graph is given by $\mathcal{G}_n = (\mathcal{V}_n, \mathcal{E}_n)$, where $\mathcal{V}_n = \mathcal{V}_{n-1} \cup \{v_n\}$ and $\mathcal{E}' = \mathcal{E} \setminus e_{jk} \cup \{e_{jn}, e_{\ln}, e_{nk}\}$. We relabel the vertices according to our labeling rule, such that the bearing Laplacian matrix is verified to be lower triangular. We verify that $\text{card}(\mathcal{P}_i) = 2$ is guaranteed $\forall 3 \leq i \leq n$. It follows from Lemma 3 that $\mathcal{G}_n$ is GBR-BP. ■

The constructed framework can be considered centralized, in the sense that two leaders have the ability to indirectly control the whole swarm. It can also be considered decentralized because each follower only needs the local information associated with its parents. This reflects the targeted MNSs concept [1], supporting parallel processing even in large-scale robot swarms.

Our proposed construction process contributes frameworks that exhibit the following key properties.

1) The framework benefits from rigidity and hierarchy. These two features provide a theoretical basis to predict the motion of each robot and can facilitate human operators controlling the behavior of the swarm.
2) On the basis of rigidity and hierarchy, we can dynamically change the size of the framework via the framework reconstruction strategies proposed in Section V. This flexibility of swarm size enables regulation of frameworks according to task requirements and environment constraints.
3) The framework has no reliance on external position or distance measurements, instead using only bearing measurement. When coupled with an example control law, e.g., for reactive formation control with moving leaders, the robot swarm can achieve self-organized formations using only relative bearing measurement and local interactions, as shown in Section VI.

Remark 1: The concept of hierarchy has been reflected in the field of bearing-based formation control, in formation maneuvering [27] and the Henneberg Construction [26]. However, our research differs from these and contributes in two major ways.

1) Bearing rigidity was not discussed in [27]. We address this gap by analyzing the rigidity of hierarchical frameworks based on the notion of bearing persistence.
2) Hierarchy was introduced as a concept in [26], but not fully investigated. We expand on the existing work and propose self-reconfiguration algorithms on the basis of hierarchical structures.

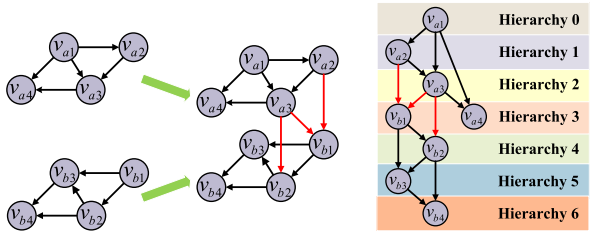

Fig. 5. Example of merging two frameworks. (a) Cube framework is constructed by merging two square frameworks (added edges in red). (b) Hierarchy layers of the post-merged framework.

V. Framework Reconstruction

In this section, we address the problem of framework reconstruction. The case of adding a new robot can be addressed directly by the vertex addition and edge splitting operations introduced in Section IV. The remaining cases of reconstruction are more challenging and require explication. Accordingly, in this section, we address the cases of *merging frameworks*, *robot departure*, and *splitting frameworks*.

A. Merging Frameworks

This section concerns the problem of merging two frameworks. A merging strategy for undirected frameworks that consider the maintenance of bearing rigidity has been proposed in [30]. We build upon [30] by extending to the case of directed graphs and maintenance of the hierarchical structure and bearing persistence.

First, consider two directed hierarchical frameworks: $(\mathcal{G}_a, p_a)$ with n_a robots, and $(\mathcal{G}_b, p_b)$ with n_b robots, where $n_a, n_b \geq 2$. Fundamentally, we need to find the minimum number of new edges to be added, in order to maintain bearing rigidity and persistence.

Theorem 2: Consider two graphs $\mathcal{G}_a = (\mathcal{V}_a, \mathcal{E}_a)$ and $\mathcal{G}_b = (\mathcal{V}_b, \mathcal{E}_b)$, generated by HHC. Denote two leaders of framework $(\mathcal{G}_b, p_b)$ as v_{b_1} and v_{b_2} and perform the following sequence of operations: 1) Select two vertices $v_{a_i}, v_{a_j} \in \mathcal{V}_a$ and 2) Add three edges $e_1 = (v_{a_i}, v_{b_1})$, $e_2 = (v_{a_j}, v_{b_1})$, and $e_3 = (v_{a_j}, v_{b_2})$. The resulting post-merged graph $\bar{\mathcal{G}} = \{\bar{\mathcal{V}}, \bar{\mathcal{E}}\}$ defined by $\bar{\mathcal{V}} = \mathcal{V}_a \cup \mathcal{V}_b$ and $\bar{\mathcal{E}} = \mathcal{E}_a \cup \mathcal{E}_b \cup \{e_1, e_2, e_3\}$ is GBR-BP.

Proof: We add two edges to v_{b_1} and one edge to v_{b_2}, which results in $\text{card}(\mathcal{P}_i) = 2$ for all $3 \leq i \leq n_a + n_b$. We can, therefore, employ Lemma 4 to verify that the post-merged graph is GBR-BP. ■

Theorem 2 implies that, after adding three edges, the post-merged graph is GBR-BP. In addition, the hierarchical structure of the framework is preserved. After the merging operation, the hierarchy of robots in the framework $(\mathcal{G}_B, p_B)$ should be updated as $h(v_{B_i}) \leftarrow h(v_{B_i}) + \max(h(v_{a_i}), (v_{a_j})) + 1$. An example of merging two frameworks is given in Fig. 5.

Motivated by Theorem 2, we extend the merging strategy to the case of m graphs, as summarized in Algorithm 2. It should be noted that using Algorithm 2, merging processes can be performed in series or in parallel. In the case of m graphs, the merging processes can be grouped into a minimum of 1 groups (i.e., all in parallel) and a maximum of $m-1$ groups (i.e., all in

Algorithm 1 Constructing a Hierarchical GBR-BP Graph of n $(n \geq 2)$ Robots in $\mathbb{R}^d$ $(d \geq 2)$

$i \leftarrow 0$;
Choose arbitrarily two robots from the swarm to define as leaders v_1 and v_2;
$\mathcal{G} \leftarrow$ add vertices v_1 and v_2, and edge $e_{2,1} = (v_2, v_1)$;
$i \leftarrow i + 2$;
$h(v_1) \leftarrow 0$, $h(v_2) \leftarrow 1$;
while $i \leq n$ **do**
  $i \leftarrow i + 1$;
  if Vertex addition is performed **then**
    Choose arbitrarily two robots to define as v_j and v_k from $\mathcal{G}$;
    $\mathcal{G} \leftarrow$ Add a vertex v_i and two edges e_{ij} and e_{ik};
    $h(v_i) \leftarrow \max\big(h(v_j)\big), h(v_k))\big) + 1$;
  else if Edge splitting is performed **then**
    Choose arbitrarily one robot to define as v_k from $\mathcal{G}$, which has two parent robots v_j and v_p;
    Choose arbitrarily one robot to define as v_l from $\mathcal{G}$, satisfying $h(v_l) \leq h(v_k)$;
    $\mathcal{G} \leftarrow$ Remove edge e_{kj}, add one robot v_i and three edges e_{ki}, e_{ij} and e_{il};
    $h(v_i) \leftarrow \max\big(h(v_j), h(v_l)\big) + 1$;
    $h(v_k) \leftarrow \max\big(h(v_i), h\big(v_p\big)\big) + 1$;
  end if
end while
return $\mathcal{G}$;

Algorithm 2 Merging m GBR-BP Graphs $\mathcal{G}_1, \mathcal{G}_2, \ldots, \mathcal{G}_m$ Into One GBR-BP Graph $\bar{\mathcal{G}}$ in $\mathbb{R}^d$ $(d \geq 2)$

$\bar{\mathcal{G}} \leftarrow \mathcal{G}_1$;
for $k = 2 \rightarrow m$ **do**
  Select two vertices v_i and v_j from $\bar{\mathcal{G}}_{k-1}$;
  Select leader vertices v_{k_1} and v_{k_2} from $\mathcal{G}_k$;
  Add edges $e_1 = \big(v_i, v_{k_1}\big)$, $e_2 = \big(v_j, v_{k_2}\big)$, and $e_3 = \big(v_j, v_{k_2}\big)$;
  $\bar{\mathcal{G}} = \big(\bar{\mathcal{V}}, \bar{\mathcal{E}}\big)$, where $\bar{\mathcal{V}} \leftarrow \bar{\mathcal{V}} \cup \mathcal{V}_k$ and $\bar{\mathcal{E}} \leftarrow \bar{\mathcal{E}} \cup \mathcal{E}_k \cup \{e_1, e_2, e_3\}$;
  Update the hierarchy of vertices in $\mathcal{G}_k$ as $h\big(v_{k_i}\big) \leftarrow h\big(v_{k_i}\big) + \max\big(h(v_i), \big(v_j\big)\big) + 1$;
end for
return $\bar{\mathcal{G}}$;

series), such that the time complexity will be between $\mathcal{O}(1)$ and $\mathcal{O}(m)$. Therefore, with the proposed merging operation, we can accelerate the construction process of large-scale robot swarms. For instance, we can construct various hierarchical and rigid frameworks simultaneously via Algorithm 1, and at the same time, Algorithm 2 can be utilized to merge these frameworks, achieving a faster self-organization process via parallelization.

B. Robot Departure

In this section, we consider the removal of a robot from the framework. According to whether a robot has a child or not, the robots in the swarm can be classified into two categories: 1) outer node (i.e., no child) and 2) inner node (i.e., at least one child). We consider the robot departure problem in both cases.

Case 1 (Removal of an Outer Node): We first consider the case with an outer node robot, e.g., v_7 and v_8 in Fig. 6(a). Consider a directed hierarchical framework $(\mathcal{G}, p)$ with n robots $(n \geq 2)$. We assume that the vertex v_n is an outer node and its parent vertices are relabeled as v_i and v_j.

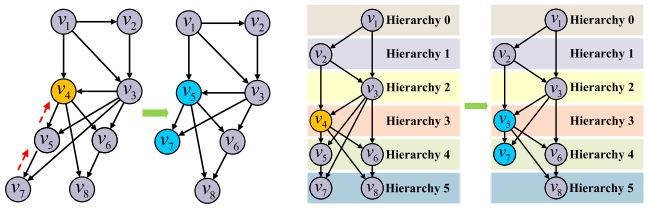

Fig. 6. Example of robot departure when using Algorithm 3. Step 1: Vertex v_4 (yellow) is removed from the framework. Step 2: Vertex v_5 (blue) is shifted to replace v_4 and vertex v_7 (blue) is shifted to replace v_5. (a) and (b) show the frameworks and the corresponding hierarchy layers, respectively, of the two steps.

Algorithm 3 Removal of Vertex v_k From GBR-BP Graph $\mathcal{G}$ in $\mathbb{R}^d$ $(d \geq 2)$

$\mathcal{V} \leftarrow \mathcal{V} \backslash \{v_k\}$, $\mathcal{E} \leftarrow \mathcal{E} \backslash \big\{e_{ij} | v_i \text{ or } v_j = v_k\big\}$;
while v_k is not an outer node vertex **do**
  Select vertex $v_m \in \mathcal{C}_k$, such that $\forall v_j \in C_k, h(v_m) \geq h\big(v_j\big)$;
  $\mathcal{G} \leftarrow$ Remove the edges associated with v_m;
  $\mathcal{G} \leftarrow$ Add edges from vertices in $\mathcal{P}_k$ to v_m;
  $\mathcal{G} \leftarrow$ Add edges from v_m to $v_j \in \mathcal{C}_k \backslash v_m$;
  $h(v_m) \leftarrow h(v_k)$;
  $v_k \leftarrow v_m$;
end while
return $\mathcal{G}$;

Theorem 3: Given a GBR-BP graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ generated by HHC, remove an outer node vertex v_n and two associated edges e_{in} and e_{jn}, the graph $\mathcal{G}^- = (\mathcal{V}^-, \mathcal{E}^-)$ defined by $\mathcal{V}^- = \mathcal{V} \backslash \{v_n\}$ and $\mathcal{E}^- = \mathcal{E} \backslash \{e_{in}, e_{jn}\}$ is GBR-BP.

The proof of Theorem 3 is omitted, here, because the removal of an outer node is an inverse operation of "vertex addition" in HHC. Lemma 3 can be used to verify the rigidity of the framework after the deletion of an outer node.

Case 2 (Removal of an Inner Node): When an inner node robot leaves the framework [e.g., v_4 in Fig. 6(a)], the rigidity of the framework is destroyed and needs to be repaired. To repair the rigidity and maintain the hierarchical structure, the following corollary can be derived.

Corollary 1: Given a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ generated by HHC, if an inner vertex v_k leaves the framework, let its position in the framework (including hierarchy and connected edges) be inherited by one of its children vertices v_m that has the highest hierarchy in $\mathcal{C}_k$. In other words $\forall v_j \in \mathcal{C}_k, h(v_m) \geq h(v_j)$. If v_m is an outer node, Theorem 3 yields that the resultant graph after performing the *inheritance* operation is GBR-BP. If v_m is an inner node, we can continue performing the inheritance operation until an outer node is reached.

The strategy stated in Corollary 1 is summarized in Algorithm 3 and an example is given in Fig. 6. With the help of our proposed algorithm, we can remove any robot from the framework without destroying rigidity, persistence, and hierarchical architecture. One advantage of the proposed method is that only *local* information is required to perform the inheritance operation. The time complexity of our proposed robot departure algorithm can be calculated as $\mathcal{O}(n)$. In contrast

Algorithm 4 Removal of k ($k \geq 2$) Vertices From GBR-BP Graph $\mathcal{G}$ in $\mathbb{R}^d$ ($d \geq 2$)

$\mathcal{V} \leftarrow \mathcal{V}\backslash\{v_{m_1}, \ldots, v_{m_k}\}$, $\mathcal{E} \leftarrow \mathcal{E}\backslash\{e_{ij} | v_i \text{ or } v_j \notin \mathcal{V}\}$;
Select two vertices $v_{l_1}, v_{l_2} \in \mathcal{V}$ such that $h(v_{l_1}) \leq h(v_{l_2}) \leq h(v_i), \forall v_i \in \mathcal{V}$;
$h(v_{l_2}) \leftarrow h(v_{l_1}) + 1$;
$h(v_i) \leftarrow h(v_i) + 2, \forall v_i \in \mathcal{V}\backslash\{v_{l_1}, v_{l_2}\}$;
for $v_i \in \mathcal{V}\backslash\{v_{l_1}, v_{l_2}\}$ **do**
  while card$\{\mathcal{P}_i\} < 2$ **do**
    $\mathcal{G} \leftarrow$ add new edge e_{qi}, where $v_q \in \mathcal{V}$ is chosen such that $v_q \notin \mathcal{P}_i$ and $h(v_q) < h(v_i)$;
  end while
  Update $h(v_i)$ according to the hierarchy of its parents;
end for
return $\mathcal{G}$;

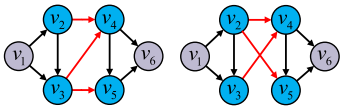

Fig. 7. Two possible configurations of directed Z-links (marked in red), based on undirected Z-links [33].

to our approach, existing methods such as [32] require an optimal repairing solution from the *global* perspective to find the necessary edges to maintain the rigidity.

Remark 2: Note that for the case of multiple robots being removed at the same time, it might not always be possible to employ the inheriting operation as introduced in Corollary 1, because the hierarchical structure would not always be maintained. To reconstruct frameworks under such a scenario, based on Lemma 4, each follower should possess at least two parent vertices. For this reason, Algorithm 4 presents a protocol for the followers of removed robots to select new parents under the hierarchy constraint.

C. Splitting Frameworks

In this section, we consider the case where a framework with at least four robots is split into several disjoint subframeworks, each consisting of at least two robots. Similar to the merging operation, the main difficulty of the splitting operation is the preservation of the bearing rigidity, persistence, and hierarchy of the subframeworks after splitting. Note that the splitting operation can be considered a generalized extension of robot departure. Without loss of generality, we first consider the strategy for splitting one framework into two subframeworks.

We use a special graph called Z-link, originally proposed in [39] and employed in [33] for undirected graphs. We extend this existing research to directed Z-links. We denote Z-link by $Z = (\mathcal{V}_Z, \mathcal{E}_Z)$, where $|\mathcal{V}_Z| = 4$ and $|\mathcal{E}_Z| = 3$, as shown in Fig. 7. The following definition determines the existence of a Z-link in a graph $\mathcal{G}$.

Definition 4 (Z-link): Consider a directed graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$. Two disjoint subgraphs $\mathcal{G}_a = (\mathcal{V}_a, \mathcal{E}_a)$ and $\mathcal{G}_b = (\mathcal{V}_b, \mathcal{E}_b)$ are said to be connected via a Z-link if the following two conditions hold.

1) There exists four distinct vertices $v_{a1}, v_{a2} \in \mathcal{V}_a$ and $v_{b1}, v_{b2} \in \mathcal{V}_b$, such that the graph among these four vertices is a Z-link.
2) $\mathcal{V}_a \cup \mathcal{V}_b = \mathcal{V}$, $\mathcal{V}_a \cap \mathcal{V}_b = \varnothing$, $\mathcal{E}_a \cap \mathcal{E}_b = \varnothing$, and $\mathcal{E} = \mathcal{E}_a \cup \mathcal{E}_b \cup \mathcal{E}_Z$.

Theorem 4: Given a GBR-BP graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, let $\mathcal{G}_a = (\mathcal{V}_a, \mathcal{E}_a)$ and $\mathcal{G}_b = (\mathcal{V}_b, \mathcal{E}_b)$ be two disjoint subgraphs of $\mathcal{G}$, which are connected via a Z-link. Then, $\mathcal{G}_a$ is GBR-BP $\Leftrightarrow$ $\mathcal{G}_b$ is GBR-BP.

Proof: We only show that $\mathcal{G}_a$ is GBR-BP $\Rightarrow$ $\mathcal{G}_b$ is GBR-BP, because the reverse is the same.

Given that $\mathcal{G}$ is GBR-BP, there exists a configuration $p = [p_1^T, \ldots, p_n^T]^T \in \mathbb{R}^{dn}$, such that $(\mathcal{G}, p)$ is IBR and BP. Let $p_a = [p_1^T, \ldots, p_{n_a}^T]^T \in \mathbb{R}^{dn_a}$ and $p_b = [p_{n_a+1}^T, \ldots, p_n^T]^T \in \mathbb{R}^{d(n-n_a)}$. Let B_b be the bearing Laplacian matrix of $(\mathcal{G}_b, p_b)$.

Without loss of generality, we assume that $\mathcal{V}_a \cap \mathcal{V}_Z = \{v_{a1}, v_{a2}\}$ and $V_b \cap V_Z = \{v_{n_a+1}, v_{n_a+2}\}$. We add edge $e_{(n_a+1)(n_a+2)}$ to the graph $\mathcal{G}$. Then, the resultant graph $\mathcal{G}^+ = (\mathcal{V}, \mathcal{E} \cup e_{(n_a+1)(n_a+2)})$ is still GBR-BP under Lemma 5. Denote B^+ as the bearing Laplacian matrix of $(\mathcal{G}^+, p)$.

We augment $\mathcal{G}_a$ to $\mathcal{G}_a^+ = (\mathcal{V}_a^+, \mathcal{E}_a^+)$, defined by $\mathcal{V}_a^+ = \mathcal{V}_a \cup \{v_{n_a+1}, v_{n_a+2}\}$ and $\mathcal{E}_a^+ = \mathcal{E}_a \cup \mathcal{E}_Z \cup e_{(n_a+1)(n_a+2)}$. Denote B_a^+ as the bearing Laplacian matrix of $(\mathcal{G}_a^+, p_a^+)$, where $p_a^+ = [p_a^T, p_{n_a+1}^T, p_{n_a+2}^T]^T \in \mathbb{R}^{d(n_a+2)}$.

As a result, we can write the bearing rigidity matrix B^+ as

$$B^+ = \begin{bmatrix} B_a^+ & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} + \begin{bmatrix} \mathbf{0} & \mathbf{0} \\ \mathbf{0} & B_b \end{bmatrix}. \tag{5}$$

Consider equation $B_b q = \mathbf{0}$. If $\mathcal{G}_b$ is not GBR-BP, then there exists a configuration $q_b = [q_{n_a+1}^T, \ldots, q_n^T] \in \mathbb{R}^{d(n-n_a)}$ such that $q_{n_a+1} = p_{n_a+1}$ and $q_{n_a+2} = p_{n_a+2}$, but $q_i \neq p_i \quad \forall i \in \{n_a + 3, \ldots, n\}$.

Let $q' = [p_1^T, \ldots, p_{n_a+1}^T, p_{n_a+2}^T, q_{n_a+3} \ldots, q_n]^T$. Equation (5) yields $B^+ q' = 0$. Note that $q' \notin \text{span}\{\mathbf{1}_n \otimes \mathbf{I}_d, p\}$, i.e., $\mathcal{G}^+$ is not GBR-BP, which is a contradiction. Therefore, $\mathcal{G}_b$ is verified to be GBR-BP. ■

Theorem 4 indicates that, for any GBR-BP graph, if there exists a Z-link connecting two disjoint subgraphs $\mathcal{G}_a$ and $\mathcal{G}_b$, one of which is guaranteed to be GBR-BP, then the other subgraph is also GBR-BP. This lemma leads us to develop the following splitting strategy: we first find a GBR-BP subgraph $\mathcal{G}_a$, and second, construct a Z-link between $\mathcal{G}_a$ and $\mathcal{G}_b$. After the removal of the Z-link edges, we obtain two GBR-BP subgraphs. We now present our two-step algorithm to split the framework, exploiting the triangularity in (4).

Step 1 (Find a GBR-BP Subgraph $\mathcal{G}_a$): Given a framework generated by HHC with n robots, the Bearing Laplacian submatrix of the first n_a robots always satisfies a triangular structure [see the triangularity in (4)]. Therefore, we can verify $\mathcal{G}_a$ as GBR-BP according to the first n_a robots, as stated in the following Theorem.

Lemma 7: Given a GBR-BP graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ generated by HHC, let $\mathcal{G}_a = (\mathcal{V}_a, \mathcal{E}_a)$ represent a subgraph describing

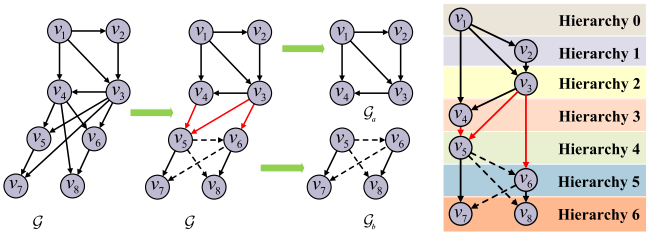

Fig. 8. Example of splitting one framework $\mathcal{G}$ into two subframeworks $\mathcal{G}_a$ and $\mathcal{G}_b$ when using Algorithm 5. (a) First, vertices v_5 and v_6 are chosen as the leaders for $\mathcal{G}_b$. Second, a Z-link (marked in red) is constructed between the two subframeworks (newly added edges are marked as dashed arrows). Third, the Z-link is removed, resulting in two separate frameworks. Corresponding hierarchy layers after Z-link construction (before Z-link removal).

Algorithm 5 Splitting one GBR-BP Graph $\mathcal{G}$ Into m GBR-BP Graphs: $\mathcal{G}_1, \mathcal{G}_2, \ldots, \mathcal{G}_m$ in $\mathbb{R}^d$ $(d \geq 2)$

$n_0 = 0$;
for $k = 1 \rightarrow m$ **do**
  $\mathcal{V}_k \leftarrow \{v_{n_{k-1}+1}, \ldots, v_{n_{k-1}+n_k}\}$;
  Select vertex $v_{n_{k-1}+1}$ as the first leader of $\mathcal{G}_k$, and denote its parents as $v_{p_k^1}$ and $v_{p_k^2}$;
  Select vertex $v_{n_{k-1}+2}$ as the second leader of $\mathcal{G}_k$;
  $\mathcal{G} \leftarrow$ Remove the ingoing edges of $v_{n_{k-1}+2}$;
  $\mathcal{G} \leftarrow$ Construct Z-link by adding two ingoing edges to the second leader: $e_{k_1} = (v_{n_{k-1}+1}, v_{n_{k-1}+2})$ and $e_{k_2} = (v_{p_k^1}, v_{n_{k-1}+2})$;
  $h(v_{n_{k-1}+2}) \leftarrow h(v_{n_{k-1}+1}) + 1$;
  $h(v_i) \leftarrow h(v_i) + 2, \forall i \in \{n_{k-1}+3, \ldots, n_{k-1}+n_k\}$;
  for $i = n_{k-1}+3$ to $n_{k-1}+n_k$ **do**
    for $v_j \in \mathcal{P}_i$ **do**
      if $v_j \notin \mathcal{V}_k$ **then**
        $\mathcal{G} \leftarrow$ remove edge e_{ji};
        $\mathcal{G} \leftarrow$ add new edge e_{qi}, where $v_q \in \mathcal{V}_k$ is chosen such that $v_q \notin \mathcal{P}_i$ and $h(v_q) < h(v_i)$;
      end if
    end for
    Update $h(v_i)$ according to the hierarchy of its parents;
  end for
  $\mathcal{E}_k \leftarrow \{e_{ij} \subset \mathcal{E} | v_i, v_j \in \mathcal{V}_k\}$;
end for
$\mathcal{G} \leftarrow$ Remove all constructed Z-links by deleting two ingoing edges of $v_{n_{k-1}+1}$ and edge e_{k_2};
return $\mathcal{G}_1, \mathcal{G}_2, \ldots, \mathcal{G}_m$;

the interactions corresponding to first n_a vertices, i.e., $\mathcal{V}_a = \{v_1, \ldots, v_{n_a}\}$. Then, $\mathcal{G}_a$ is GBR-BP.

Proof: Denote B as the bearing rigidity Laplacian matrix of $(\mathcal{G}, p)$, where $p \in \mathbb{R}^{nd}$ is a configuration. Then, B can be partitioned as

$$B = \begin{bmatrix} B_a & \mathbf{0} \\ B_{b1} & B_{b2} \end{bmatrix} \tag{6}$$

where $B_a \in \mathbb{R}^{dn_a \times dn_a}$ denotes the bearing Laplacian matrix for the first n_a vertices. Then, we can apply statement (5) of Lemma 3 to verify the rank of matrices on diagonal of B_a, which shows that $\mathcal{G}_a$ is GBR-BP. ■

Step 2 (Construct a Z-Link Between Two Subgraphs): Let $\mathcal{G}_b = (\mathcal{V}_b, \mathcal{E}_b)$ represent the interactions among the remaining vertices, i.e., $\mathcal{V}_b = \{v_{n_a+1}, \ldots, v_n\}$. Corresponding to Definition 4, Z-link construction comprises the following two parts.

1) Let $\mathcal{P}_{n_a+1} = \{v_{p1}, v_{p2}\}$. Remove the ingoing edges of v_{n_a+2}, then add edges $e_{(n_a+1)(n_a+2)}$ and $e_{p_1(n_a+2)}$.
2) $\forall v_i \in V_b \backslash \{v_{n_a+1}, v_{n_a+2}\}$, if its parent $v_j \in V_a$, then remove e_{ji}. To preserve rigidity, add new edge e_{ki}, where new parent $v_k \in \mathcal{V}_b$ is chosen such that $v_k \notin \mathcal{P}_i$ and $h(v_k) < h(v_i)$.

Here, Lemma 6 is repeatedly employed to satisfy Definition 4, therefore, the resultant graph is still GBR-BP.

Finally, we can use the above splitting strategy for the case of m graphs, as summarized in Algorithm 5. Similar to the merging process, according to the for-loop structure of the splitting algorithm, the time complexity can be calculated as $\mathcal{O}(mn)$. Note that multiple splitting processes can happen in parallel. This is because the robots have been divided into m subsets $\mathcal{V}_k = \{v_{n_{k-1}+1}, \ldots, v_{n_{k-1}+n_k}\}$, and each follower is required to change its parent vertex such that $\forall v_i \in \mathcal{V}_k, \mathcal{P}_i \subset \mathcal{V}_k$. For instance, we can simultaneously apply the splitting algorithm to all subsets m; then the time complexity will be determined by the maximum subset size, so it will be reduced to $\mathcal{O}(\max\{n_k\})$. Therefore, in practice, the time complexity will often be lower than $\mathcal{O}(mn)$. A splitting example is presented in Fig. 8. Note that Algorithm 5 can split the framework into at least two subgraphs with arbitrary size no less than 2, which provides flexibility in managing the size of the framework.

VI. Validation With Example Control Law

In this section, to demonstrate and validate our theoretical results, we couple our proposed hierarchical frameworks with an example formation control law and then apply them to four example scenarios of reactive formation control in an aerial robot swarm with moving leaders. In the first scenario, we establish a target formation based on our proposed hierarchical framework and validate Theorem 1. In the second, we merge two formations under Algorithm 2 and validate Theorem 2. Third, we show robot departure from a formation under Algorithm 3 and validate Theorem 3 and Corollary 1. Fourth, we split a formation under Algorithm 5 and validate Theorem 4 and Lemma 7.

We consider a group of n mobile robots moving in $\mathbb{R}^d$ $(d \geq 2)$, the model of which is described by a single integrator $\dot{p}_i = u_i$, where $p_i \in \mathbb{R}^d$ is the inertial position of the ith robot and $u_i \in \mathbb{R}^d$ is the control input. The main purpose of this section is to validate our proposed construction and reconstruction algorithms, when coupled with an example control law, using experimental results in simulation. Therefore, only a single-integrator model is considered. (For further results on formation control with two-leader directed frameworks, please refer to [22] and [23].)

A. Example Scenario 1: Achieving the Desired Formation

Consider a swarm with n robots with a hierarchical topology, characterized by a directed graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ generated by Algorithm 1. The robots are located at $p_1, \ldots, p_n$ in $\mathbb{R}^d$, and each robot does not know the global position p_i but can

sense the bearing vectors with regard to its parent robots, i.e., $\{g_{ij}|v_j \in \mathcal{P}_i\}$. We assume that the positions of the neighboring robots do not coincide, i.e., $\forall e_{ij} \in \mathcal{E}$ and $t \geq 0$, $p_i(t) \neq p_j(t)$, which guarantees the bearing vectors to be well defined.

In our hierarchical framework, two robots denoted by v_1 and v_2 are chosen as the leaders, while the others are followers. Given a set of feasible desired bearings $\{g^*_{ij}|e_{ji} \in \mathcal{E}\}$ among the robots,[2] the desired robot formation is uniquely characterized, but for rigid translation and scaling. This last ambiguity of the graph is resolved by fixing the position of the first robot, and the distance of the first two robots. Define $p^*(t)$ as the vector of the desired position of the robots over time, and $d^*_{ij}(t) = ||p^*_i(t) - p^*_j(t)||$ as the distance between the desired position of robot i and of robot j. The relationship between $p^*(t)$ and $\{g^*_{ij}|e_{ji} \in \mathcal{E}\}$ is characterized by [26, Lemma 1].

According to [26, Lemma 1], given the position of two leaders, a framework constructed by HHC can be uniquely determined. Moreover, the desired translational and scaling maneuvers of the formation are uniquely determined by the reference motion of the first two "leader robots." This inherent property also shows the possibility of using *centralized* decision-making behaviors with our self-organized hierarchical frameworks, because we can control the translation and scale of the formation via two leader robots, which reduces the complexity of formation management.

In this article, for the sake of simplicity, we do not consider the motion control of leaders, and we assume that the two leaders move along predefined trajectories, i.e., $p_1(t) = p^*_1(t)$ and $p_2(t) = p_1(t) - d^*_{2,1}(t)g^*_{2,1}$, at all time $t > 0$.[3] In order to drive the followers to achieve the desired formation, the following bearing-only formation control law for robot v_i $(i \geq 3)$ is used:

$$\dot{p}_i = u_i = -c\left(P_{g_{ij}} g^*_{ij} + P_{g_{ik}} g^*_{ik}\right) + \dot{p}^*_i \tag{7}$$

where c is a positive constant to be tuned and $\dot{p}^*_i$ is a feedforward term given as follows:

$$\dot{p}^*_i(t) = \left(P_{g^*_{ij}} + P_{g^*_{ik}}\right)^{-1}\left(P_{g^*_{ij}}\dot{p}^*_j(t) + P_{g^*_{ik}}\dot{p}^*_k(t)\right) \tag{8}$$

and each agent can compute $\dot{p}^*_i(t)$ by receiving $\dot{p}^*_j(t)$ and $\dot{p}^*_k(t)$ from its parents.

Remark 3: The control law (7) is inspired by [26], in which the formation is static. To extend this zero-velocity control law to moving formations, we introduce the feedforward term $\dot{p}^*_i(t)$ to guarantee zero steady-state error. Note that transmission and computation of feedforward terms through the hierarchy is not instantaneous and will introduce delays. There are several approaches to handle such delays (e.g., sufficient preview of the reference signal, or relaxing the perfect tracking requirement and proving ISS-like properties assuming a purely reactive control law). However, such analyses are nontrivial and are beyond the scope of this article. The specific control law used (7) is just an example to demonstrate the effectiveness of our proposed framework; any control law that makes use of bearing rigidity could in principle be coupled with our frameworks.

[2]The feasibility conditions are specified in [26, Assumption 2].

[3]Note that this assumption is not limitative and that all the results of this article hold true by using suitable control laws for the two leaders ensuring convergence to the desired trajectories.

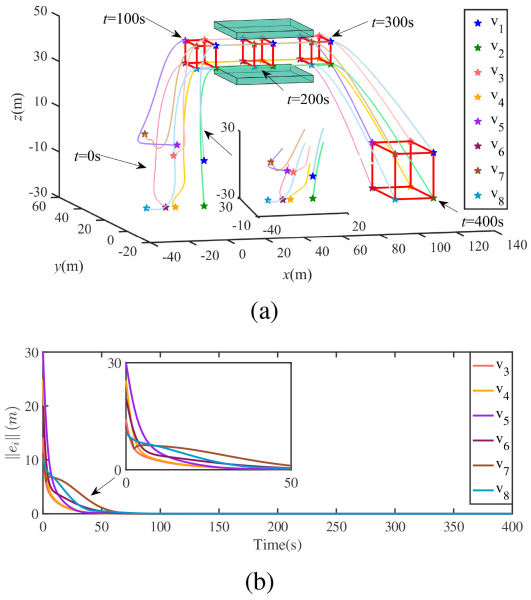

Fig. 9. Simulation results in example scenario 1: Achieving the desired formation. (a) Trajectories of robots. (b) Formation tracking errors of robots.

By employing a similar stability analysis as shown in [26, Th. 1], we can also demonstrate that the formation tracking error $e_i(t) = p(t) - p^*(t)$ asymptotically converges to zero using the control law (7). Note that the implementation of control law (7) requires only local measurement and local communication from parents to children, which supports the *decentralized* coordination targeted in a reactive swarm approach.

Remark 4: The only parameter to be tuned in (7) is the control gain c. c should be positive, and an increase of c will speed up the formation achievement, but will also result in a larger velocity amplitude. Therefore, the tradeoff between convergence speed and velocity amplitude should be considered when defining c.

A simulation example is shown in Fig. 9. We consider a group of eight robots with a hierarchical framework shown in Fig. 4(a), and the target formation is a cube. The motion of leader v_1 and time-varying distance $d^*_{2,1}$ are shown at the bottom of the page (9). The controller parameter is chosen as $c = 5$. Fig. 9(a) depicts the trajectory of eight robots.

$$\begin{cases} p_1(t) = \left[0.3t, 40\sin(\pi t/200), 40\sin(\pi t/200)\right]^T, d^*_{2,1}(t) = 20 - 10\sin(\pi t/200), t \leq 100 \\ p_1(t) = [0.3t, 40, 40]^T, d^*_{2,1}(t) = 10, 100 \leq t \leq 300 \\ p_1(t) = \left[0.3t, 40\sin(\pi(t-200)/200), 40\sin(\pi(t-200)/200)\right]^T, d^*_{2,1}(t) = 20 - 10\sin(\pi(t-200)/200), t \geq 300 \end{cases} \tag{9}$$

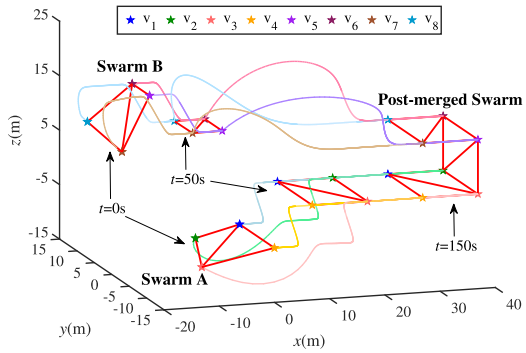

Fig. 10. Formation evolution in example scenario 2: Formation merging.

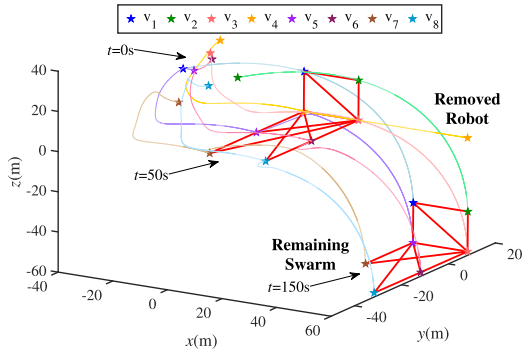

Fig. 11. Formation evolution in example scenario 3: Robot departure from the formation.

As can be seen, the target formation can be achieved while the centroid and scale of the framework is time-varying in order to pass through narrow passages. In this example, the trajectories of leaders are predefined (but known only to leaders). In practical missions, the leaders can generate trajectories in realtime according to task requirements and environment constraints. The formation tracking errors in Fig. 9(b) also converge to zero asymptotically, but with various convergence rates. Robots v_3 and v_4 have lower hierarchy and, therefore, will converge faster than the others. The simulation results in Fig. 9 validate Theorem 1 of our proposed approach.

B. Example Scenario 2: Formation Merging

Consider two robot swarms (Swarm A and Swarm B), with graphs constructed by Algorithm 1, that need to merge. Following Algorithm 2, three directed edges need to be added for the two frameworks to be merged, such that the two leaders of Swarm B become followers of Swarm A. Then, having been given a new target formation with desired bearing vectors such that the post-merged framework is IBR and BP, the robots of the post-merged swarm will move under the formation control law to achieve the target formation.

A simulation example is shown in Fig. 10. In the first 50 s, the two frameworks each achieve the target square via control law (7), but with different scales. From 50 s, following Algorithm 2, three edges are newly added to merge the two frameworks, and the target cube is achieved at 150 s. Note that the scale of Swarm B is increased to that of Swarm A after performing the merging operation, which also demonstrates that Swarm B integrates successfully into Swarm A. The simulation results in Fig. 10 validate Theorem 2 of our proposed approach.

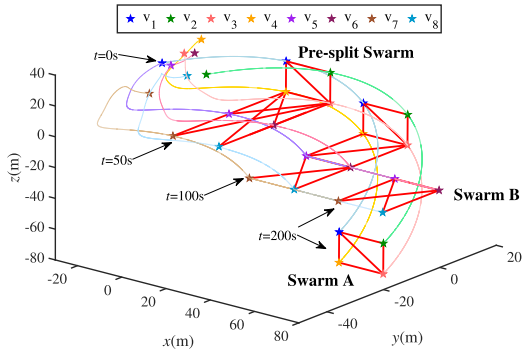

Fig. 12. Formation evolution in example scenario 4: Formation splitting.

C. Example Scenario 3: Robot Departure From Formation

In this example, the robot departure shown in Fig. 6 will be validated. In other words, robot v_4 will be removed, and then Algorithm 3 will be performed to guarantee both the hierarchy and rigidity of the framework. Fig. 11 depicts the simulation results. From 0 to 50 s, the target formation will be achieved via the control law (7). From 50 s, robot v_4 will keep moving along a straight line, and then robot v_5 will replace the position of v_4 in the framework, while v_7 will further replace the position of v_5, since v_5 is not an outer node robot. The simulation results show that the rigidity of the resulting framework is preserved because the formation is not destroyed after the removal of robot v_4, thus, validating Theorem 3 and Corollary 1 of our proposed approach.

D. Example Scenario 4: Formation Splitting

This example validates the splitting process as presented in Fig. 8(b), where a framework $(\mathcal{G}, p)$ (i.e., presplit Swarm in Fig. 12), including eight robots is split into two subframeworks $(\mathcal{G}_a, p_a)$ (i.e., Swarm A in Fig. 12) and $(\mathcal{G}_b, p_b)$ (i.e., Swarm B in Fig. 12), each with four robots.

The simulation result is given in Fig. 12. At $t = 50$ s, Z-link is constructed. From 50 s to 100 s, it can be noticed that the formation of the framework is maintained after Z-link construction, and also that the Z-link construction does not affect the rigidity of the framework. At $t = 100$ s, Z-link will be removed and two subframeworks will result. Thus, from 100 s to 200 s, robots v_5 and v_6 will be the leaders of Swarm B and the two subframeworks will move separately. The simulation results show that two subframeworks satisfy the bearing rigidity because the formations are maintained after splitting, thus, validating Theorem 4 and Lemma 7.

VII. Conclusion and Future Work

A. Conclusion

In this article, we demonstrated the construction of self-reconfigurable hierarchical frameworks for formation control

of robot swarms, based on bearing rigidity under directed topologies.

Self-organized hierarchical control had previously been identified as a promising approach to ease the design and management of collective behaviors in robot swarms [40], and hierarchical frameworks had already been demonstrated in practical studies using the MNSs paradigm [1], [12]. However, strong theoretical foundations were still needed, especially, for self-organized hierarchy to be viable for large-scale swarms of fast robots. In this article, we provided the first systematic and mathematically analyzable protocol for the implementation of self-reconfigurable hierarchical frameworks in robot swarms.

To enable self-organized hierarchical control with mathematically provable properties, we introduced a hierarchy property into the conventional Henneberg construction and extended bearing rigidity to directed graphs with lower-triangular structure. We studied self-reconfigurable hierarchical frameworks in three-key reconstruction problems: 1) merging; 2) robot departure; and 3) splitting. Finally, we demonstrated our frameworks by combining them with an example formation controller and validated our theoretical results concerning hierarchy and rigidity preservation during reconfiguration via simulation experiments in four example scenarios.

B. Future Work

This article studied bearing rigidity of directed frameworks with lower-triangular structure, which is an important initial step toward extending [25] into a general bearing rigidity theory that encompasses directed graphs. In future work, special cases of directed graphs that are IBR but do not have a lower-triangular structure, and that are relevant to formation control, would also need to be solved to complete a general bearing rigidity theory.

While in this article, we presented a simple control law, mostly for demonstrative purposes, future works could also explore new control laws for formations that are uniquely determined by two leaders and bearing constraints, to address important issues, such as navigation in obstacle-constructed environments. For instance, a protocol could be developed for formation parameters (e.g., translation, scale, and rotation) to be optimized online by two leaders, subject to obstacle constraints. As another example, a local path planner could be designed for followers to determine the interagent bearing vectors needed to avoid collisions.

APPENDIX A
PROOF OF LEMMA 3

Proof: According to the definitions of infinitesimal bearing rigidity and bearing persistence, (1) $\Leftrightarrow$ (3) $\Leftrightarrow$ (4) is straightforward. We, therefore, only show (2) $\Leftrightarrow$ (3) and (4) $\Leftrightarrow$ (5).

(2) $\Rightarrow$ (3): To demonstrate $\text{Null}(B) = \text{span}\{\mathbf{1}_n \otimes \mathbf{I}_d, p\}$, it is equivalent to show $\forall q = [q_1^T, \ldots, q_n^T]^T \in \text{Null}(B)$, $q = ap + \mathbf{1}_n \otimes b$, where $a \in \mathbb{R} \setminus \{0\}$ and $b \in \mathbb{R}^d$.

For robot 1, q_1 can be chosen randomly according to (3), thus, it is always possible to find a and b, such that $q_1 = ap_1 + b$.

For robot 2, q_2 satisfies $B_{2,1}(q_2 - q_1) = \mathbf{0}$. If $B_{2,1} = \mathbf{0}$, there is no bearing constraint for robot 2, thus, robot 2 can be placed randomly, which contradicts statement (2). Therefore, $q_2 - q_1 \in \text{Null}(B_{2,1}) = \text{span}\{p_2 - p_1\}$, i.e., $q_2 - q_1 = \alpha(p_2 - p_1)$ with $\alpha \in \mathbb{R} \setminus \{0\}$. Now, we claim that $q_i = q_1 + \alpha(p_i - p_1)$ for all $1 \le i \le n$, and use mathematical induction to check whether this claim is true.

For robot 3, the constraint is $(B_{3,1} + B_{3,2})q_3 = B_{3,1}q_1 + B_{3,2}q_2$. Using $q_2 - q_1 = \alpha(p_2 - p_1)$, the constraint can be rewritten as

$$\begin{aligned}\big(B_{3,1} + B_{3,2}\big)q_3 &= \big(B_{3,1} + B_{3,2}\big)q_1 + \alpha B_{3,2}(p_2 - p_1)\\ &= \big(B_{3,1} + B_{3,2}\big)(q_1 + \alpha(p_3 - p_1)) \end{aligned} \tag{10}$$

where the last equality uses $B_{3,1}(p_3 - p_1) + B_{3,2}(p_3 - q_2) = 0$. Under Lemma 2, $B_{3,1} + B_{3,2}$ is not singular if and only if $g_{3,1}$ and $g_{3,2}$ exist, and are not collinear. If $g_{3,1}$ (or $g_{3,2}$) does not exist, robot 3 only has one bearing constraint, thus, it has a noninfinitesimal bearing motion [such as robot 3 in Fig. 3(c)], thus, the framework cannot be uniquely determined. If $g_{3,1}$ are collinear with $g_{3,2}$, robot 3 still only has one bearing constraint, and the framework will not be unique. This implies that $B_{3,1} + B_{3,2}$ is not singular, and we obtain $q_3 = q_1 + \alpha(p_3 - p_1)$.

Now, we assume that $q_k = q_1 + \alpha(p_k - p_1)$ is true for $1 \le k \le i - 1$. For robot i, we have

$$\begin{aligned}\sum_{j=1}^{i-1} B_{ij}q_i &= \sum_{j=1}^{i-1} B_{ij}q_j\\ &= \sum_{j=1}^{i-1} B_{ij}q_1 + \alpha \sum_{j=1}^{i-1} B_{ij}\big(p_j - p_1\big)\\ &= \sum_{j=1}^{i-1} B_{ij}(q_1 + \alpha(p_i - p_1)) \end{aligned} \tag{11}$$

where the last equality uses $\sum_{j=1}^{i-1} B_{ij}(p_i - p_j) = 0$. Via similar analysis for robot 3, the uniqueness of the framework ensures that $\sum_{j=1}^{i} B_{ij}$ is nonsingular and further that $q_i = q_1 + \alpha(p_i - p_1)$.

By the above induction, we prove that $q_i = q_1 + \alpha(p_i - p_1)$ is true for all robots. Moreover, it can be derived that $q = \alpha p + \mathbf{1}_n \otimes (q_1 - \alpha p_1)$. This implies that $\forall q \in \text{Null}(B)$, $q \in \text{span}\{\mathbf{1}_n \otimes \mathbf{I}_d, p\}$.

(3) $\Rightarrow$ (2): Consider an IBR and BP framework $(\mathcal{G}, p)$. For a configuration $q \in \mathbb{R}^n$, we say q is a realization of directed graph $\mathcal{G}$, if $P_{p_i - p_j}(q_i - q_j) = 0$ for all $e_{ij} \in \mathcal{E}$. Denote the set of all realizations of $\mathcal{G}$ as $S_{\mathcal{G}}$. Our objective is to demonstrate that $\forall q \in S_{\mathcal{G}}$, $q \in \text{span}\{\mathbf{1}_n \otimes \mathbf{I}_d, p\}$. This can be directly verified via the bearing Laplacian. Given that

$$Bq = \begin{bmatrix} \vdots \\ \sum_{j=1}^{i-1} B_{ij}\big(q_i - q_j\big) \\ \vdots \end{bmatrix} = \mathbf{0} \tag{12}$$

then $q \in \text{Null(B)} = \text{span}\{\mathbf{1}_n \otimes \mathbf{I}_d, p\}$.

(4) $\Rightarrow$ (5): According to (2), $\text{rank}(B_{2,2}) = d - 1$ if $e_{12} \in \mathcal{E}$, and $\text{rank}(B_{2,2}) = 0$ otherwise. Assume $e_{12} \notin \mathcal{E}$, then p_1 and p_2 can be placed arbitrarily, which is a contradiction with statement (2). For robot i ($i \ge 3$), under Lemma 2, we assume

$\text{rank}(B_{i,i}) \neq d$, then robot i only has at most one bearing constraint, such that it can either be randomly placed in $\mathbb{R}^d$ if $\text{card}(\mathcal{P}_i) = 0$, or randomly placed along a line, if either $\text{card}(\mathcal{P}_i) = 1$ or $\forall v_j, v_k \in \mathcal{P}_i, g_{ij} = g_{ik}$. In each of these cases, the framework cannot be unique. Therefore, $\text{rank}(B_{ii}) = d$.

(5) $\Rightarrow$ (4): By the property of block matrices, $\text{rank}(B) \geq \sum_{i=2}^{n} \text{rank}(B_{ii}) = dn - d - 1$. Note that $\text{rank}(B) \leq dn - d - 1$ exists, hence, $\text{rank}(B) = dn - d - 1$. ■

APPENDIX B
PROOF OF LEMMA 4

Proof (Necessity): If $\mathcal{G}$ is GBR-BP, there exists a configuration p such that $(\mathcal{G}, p)$ is IBR and BP. Therefore, statement (5) in Lemma 3 should be satisfied. Since $\text{rank}(B_{2,2}) = d-1$, robot 1 should be the parent of robot 2. For $i \geq 3$, $\text{rank}(B_{i,i}) = d$ if and only if at least two of $\{g_{ik}\}_{k\in\mathcal{P}_i}$ are not collinear. Thus, $\text{card}(\mathcal{P}_i) \geq 2$.

Sufficiency: If $\text{card}(\mathcal{P}_2) = 1$ and $\text{card}(\mathcal{P}_i) \geq 2$ exists, we should find a configuration $p = [p_1^T, p_2^T, \ldots, p_n^T] \in \mathbb{R}^{dn}$, such that $(\mathcal{G}, p)$ is IBR and BP. For p_1, it can be selected randomly. For p_2, because robot 1 is the parent of robot 2, we only need to select $p_2 \neq p_1$, which guarantees $\text{rank}(B_{2,2}) = d - 1$. For p_i $(i \geq 3)$, because the position of its parents have been determined, p_i can be selected such that there exist at least two vertices $v_j, v_k \in \mathcal{P}_i$ with $g_{ij} \neq g_{ik}$, which guarantees $\text{rank}(B_{i,i}) = d$. In this way, we find one configuration p, such that statement 5 in Lemma 3 is satisfied. Thus, $\mathcal{G}$ is GBR-BP. ■

REFERENCES

[1] N. Mathews, A. L. Christensen, R. O'Grady, F. Mondada, and M. Dorigo, "Mergeable nervous systems for robots," *Nat. Commun.*, vol. 8, no. 1, pp. 1–7, 2017.

[2] E. Tuci, M. H. Alkilabi, and O. Akanyeti, "Cooperative object transport in multi-robot systems: A review of the state-of-the-art," *Front. Robot. AI*, vol. 5, p. 59, May 2018.

[3] C. Robin and S. Lacroix, "Multi-robot target detection and tracking: Taxonomy and survey," *Auton. Robot.*, vol. 40, no. 4, pp. 729–760, 2016.

[4] M. Dorigo et al., "Evolving self-organizing behaviors for a swarm-bot," *Auton. Robot.*, vol. 17, nos. 2–3, pp. 223–245, 2004.

[5] S. Nouyan, R. GroSS, M. Bonani, F. Mondada, and M. Dorigo, "Teamwork in self-organized robot colonies," *IEEE Trans. Evol. Comput.*, vol. 13, no. 4, pp. 695–711, Aug. 2009.

[6] M. Dorigo et al., "Swarmanoid: A novel concept for the study of heterogeneous robotic swarms," *IEEE Robot. Autom. Mag.*, vol. 20, no. 4, pp. 60–71, Dec. 2013.

[7] J. Werfel, K. Petersen, and R. Nagpal, "Designing collective behavior in a termite-inspired robot construction team," *Science*, vol. 343, no. 6172, pp. 754–758, 2014.

[8] M. Rubenstein, A. Cornejo, and R. Nagpal, "Programmable self-assembly in a thousand-robot swarm," *Science*, vol. 345, no. 6198, pp. 795–799, 2014.

[9] G. Li, D. St-Onge, C. Pinciroli, A. Gasparri, E. Garone, and G. Beltrame, "Decentralized progressive shape formation with robot swarms," *Auton. Robot.*, vol. 43, no. 6, pp. 1505–1521, 2019.

[10] F. Schiano, A. Franchi, D. Zelazo, and P. R. Giordano, "A rigidity-based decentralized bearing formation controller for groups of quadrotor UAVs," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, 2016, pp. 5099–5106.

[11] K.-K. Oh, M.-C. Park, and H.-S. Ahn, "A survey of multi-agent formation control," *Automatica*, vol. 53, pp. 424–440, Mar. 2015.

[12] W. Zhu, M. Allwright, M. K. Heinrich, S. Oğuz, A. L. Christensen, and M. Dorigo, "Formation control of UAVs and mobile robots using self-organized communication topologies," in *Proc. 12th Ants Conf.*, 2020, pp. 306–314.

[13] A. Jamshidpey, W. Zhu, M. Wahby, M. Allwright, M. K. Heinrich, and M. Dorigo, "Multi-robot coverage using self-organized networks for central coordination," in *Proc. 12th Ants Conf.*, 2020, pp. 306–314.

[14] Q. Yang, M. Cao, and B. D. Anderson, "Growing super stable tensegrity frameworks," *IEEE Trans. Cybern.*, vol. 49, no. 7, pp. 2524–2535, Jul. 2019.

[15] F. Mehdifar, F. Hashemzadeh, M. Baradarannia, and M. de Queiroz, "Finite-time rigidity-based formation maneuvering of multiagent systems using distributed finite-time velocity estimators," *IEEE Trans. Cybern.*, vol. 49, no. 12, pp. 4473–4484, Dec. 2019.

[16] S. Zhao and D. Zelazo, "Translational and scaling formation maneuver control via a bearing-based approach," *IEEE Trans. Control Netw. Syst.*, vol. 4, no. 3, pp. 429–438, Jun. 2015.

[17] S. Zhao and D. Zelazo, "Bearing rigidity and almost global bearing-only formation stabilization," *IEEE Trans. Autom. Control*, vol. 61, no. 5, pp. 1255–1268, May 2016.

[18] X. Li, C. Wen, and C. Chen, "Adaptive formation control of networked robotic systems with bearing-only measurements," *IEEE Trans. Cybern.*, vol. 51, no. 1, pp. 199–209, Jan. 2021.

[19] X. Li, C. Wen, X. Fang, and J. Wang, "Adaptive bearing-only formation tracking control for nonholonomic multiagent systems," *IEEE Trans. Cybern.*, vol. 52, no. 8, pp. 7552–7562, Aug. 2022, doi: 10.1109/TCYB.2020.3042491.

[20] Z. Li, H. Tnunay, S. Zhao, W. Meng, S. Q. Xie, and Z. Ding, "Bearing-only formation control with prespecified convergence time," *IEEE Trans. Cybern.*, vol. 52, no. 1, pp. 620–629, Jan. 2022, doi: 10.1109/TCYB.2020.2980963.

[21] J. Zhao, X. Li, X. Yu, and H. Wang, "Finite-time cooperative control for bearing-defined leader-following formation of multiple double-integrators," *IEEE Trans. Cybern.*, vol. 52, no. 12, pp. 13363–13372, Aug. 2022.

[22] Y. Zhang, X. Wang, S. Wang, and X. Tian, "Distributed bearing-based formation control of unmanned aerial vehicle swarm via global orientation estimation," *Chin. J. Aeronaut.*, vol. 35, no. 1, pp. 44–58, 2022.

[23] Y. Zhang, S. Wang, X. Wang, and X. Tian, "Bearing-based formation control for multiple underactuated autonomous surface vehicles with flexible size scaling," *Ocean Eng.*, vol. 267, Jan. 2023, Art. no. 113242.

[24] F. Arrigoni and A. Fusiello, "Bearing-based network localizability: A unifying view," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 41, no. 9, pp. 2049–2069, Sep. 2018.

[25] S. Zhao and D. Zelazo, "Bearing rigidity theory and its applications for control and estimation of network systems: Life beyond distance rigidity," *IEEE Control Syst. Mag.*, vol. 39, no. 2, pp. 66–83, Apr. 2019.

[26] M. H. Trinh, S. Zhao, Z. Sun, D. Zelazo, B. D. Anderson, and H.-S. Ahn, "Bearing-based formation control of a group of agents with leader-first follower structure," *IEEE Trans. Autom. Control*, vol. 64, no. 2, pp. 598–613, Feb. 2019.

[27] M. H. Trinh and H.-S. Ahn, "Finite-time bearing-based maneuver of acyclic leader–follower formations," *IEEE Control Syst. Lett.*, vol. 6, pp. 1004–1009, 2021.

[28] T.-S. Tay and W. Whiteley, "Generating isostatic frameworks," *Struct. Topol.*, vol. 11, pp. 21–69, Jan. 1985.

[29] T. Eren, "Formation shape control based on bearing rigidity," *Int. J. Control*, vol. 85, no. 9, pp. 1361–1379, 2012.

[30] M. H. Trinh, Q. Van Tran, and H.-S. Ahn, "Minimal and redundant bearing rigidity: Conditions and applications," *IEEE Trans. Autom. Control*, vol. 65, no. 10, pp. 4186–4200, Oct. 2020.

[31] A. Karimian and R. Tron, "Theory and methods for bearing rigidity recovery," in *Proc. 56th Conf. Decis. Control*, Dec. 2017, pp. 2228–2235.

[32] Y. Hou and C. Yu, "Elementary operations for rigidity restoration and persistence analysis of multi-agent system," *IET Control Theory Appl.*, vol. 10, no. 2, pp. 119–125, 2016.

[33] D. Carboni, R. K. Williams, A. Gasparri, G. Ulivi, and G. S. Sukhatme, "Rigidity-preserving team partitions in multiagent networks," *IEEE Trans. Cybern.*, vol. 45, no. 12, pp. 2640–2653, Dec. 2015.

[34] S. Zhao and D. Zelazo, "Bearing-based formation stabilization with directed interaction topologies," in *Proc. 55th Conf. Decis. Control*, Osaka, Japan, Dec. 2015, pp. 6115–6120.

[35] J. M. Hendrickx, B. D. Anderson, J.-C. Delvenne, and V. D. Blondel, "Directed graphs for the analysis of rigidity and persistence in autonomous agent systems," *Int. J. Robust Nonlinear Control*, vol. 17, no. 10-11, pp. 960–981, 2007.

[36] S. Zhao, Z. Sun, D. Zelazo, M.-H. Trinh, and H.-S. Ahn, "Laman graphs are generically bearing rigid in arbitrary dimensions," in *Proc. 56th Conf. Decis. Control*, Dec. 2017, pp. 3356–3361.

[37] M. Nagy, Z. Ákos, D. Biro, and T. Vicsek, "Hierarchical group dynamics in pigeon flocks," *Nature*, vol. 464, no. 7290, pp. 890–893, 2010.
[38] G. Laman, "On graphs and rigidity of plane skeletal structures," *J. Eng. Math.*, vol. 4, no. 4, pp. 331–340, 1970.
[39] R. Olfati-Saber and R. Murray, "Graph rigidity and distributed formation stabilization of multi-vehicle systems," in *Proc. 41st Conf. Decis. Control*, Las Vegas, NV, USA, Dec. 2002, pp. 2965–2971.
[40] M. Dorigo, G. Theraulaz, and V. Trianni, "Reflections on the future of swarm robotics," *Sci. Robot.*, vol. 5, no. 49, 2020, Art. no. eabe4385.

Yuwei Zhang received the B.S. degree in mathematics and applied mathematics and the Ph.D. degree in mechanical engineering from Beihang University, Beijing, China, in 2016 and 2021, respectively.

From 2019 to 2021, he was a Visiting Research Student with IRIDIA, the Artificial Intelligence Lab, Université Libre de Bruxelles, Brussels, Belgium. He is currently a Postdoctoral Researcher with the School of Automation Science and Electrical Engineering, Beihang University. His research interests include nonlinear control of marine vehicles and cooperative control of robot swarm.

Sinan Oğuz received the bachelor's degree in electrical and electronics engineering from Ankara University, Ankara, Turkey, in 2017. Since October 2019, he has been pursuing the Ph.D. degree with IRIDIA, the Artificial Intelligence Laboratory, and SAAS, the Department of Control Design and Systems Analysis, Université Libre de Bruxelles, Brussels, Belgium.

From September 2015 to April 2017, he worked on Passive Radar Systems and Radar Tracking Algorithms, Ankara University Estimation and Fusion Laboratory with the Scientific and Technological Research Council of Turkey (TÜBİTAK) Undergraduate Research Fellowship. From April 2017 to February 2018, he worked as a Full-Time Avionics Software Engineer with ASELSAN Inc., Ankara, where he designed and programmed sensor fusion and state estimation algorithms for aircraft navigation systems. From February 2018 to October 2019, he worked as a Full-Time Research Engineer with STM Defense Inc., Ankara, on autonomous UAVs, swarm intelligence, and GPS-less navigation. His primary research interests include swarm robotics, networked control systems, self-organizing networks, GPS-less navigation, and UAVs.

Shaoping Wang received the B.Eng., M.Eng., and Ph.D. degrees in mechatronics engineering from Beihang University, Beijing, China, in 1988, 1991, and 1994.

She has been with the School of Automation Science and Electrical Engineering, Beihang University since 1994 and promoted to the Rank of Professor in 2000. Her research interests include engineering reliability, fault diagnostic, prognostic, health management, and fault-tolerant control.

Dr. Wang was honored as a Changjiang Scholar Professor by the Ministry of Education of China in 2013.

Emanuele Garone (Member, IEEE) received the master's degree *(Laurea)* and the Ph.D. degree in systems engineering from the University of Calabria, Rende, Italy, in 2005 and 2009, respectively.

Since 2010, he is an Associate Professor with the Control Design and System Analysis Department, Université Libre de Bruxelles, Brussels, Belgium. He has authored more than 160 contributions in peer-reviewed conferences and journals. His interests include constrained control, reference governors, networked control systems, and aerial robotics.

In 2014, Dr. Garone received an Honorable Mention for the Young Author Prize of the 19th IFAC World Congress.

Xingjian Wang received the B.Eng. and Ph.D. degrees in mechatronics engineering from Beihang University, Beijing, China, in 2006 and 2012, respectively.

From 2009 to 2010, he was a Visiting Scholar with the School of Mechanical Engineering, Purdue University, West Lafayette, IN, USA. He is currently with the School of Automation Science and Electrical Engineering, Beihang University. His research interests include adaptive and nonlinear control, fault diagnosis, prognostic, health management, and fault-tolerant control.

Marco Dorigo (Fellow, IEEE) received the Master of Technology degree *(Laurea)* in industrial technologies engineering and the Ph.D. degree in electronic engineering from the Politecnico di Milano, Milan, Italy, in 1986 and 1992, respectively, and the title of Agrégé de l'Enseignement Supérieur from Université Libre de Bruxelles (ULB), Brussels, Belgium, in 1995.

From 1992 to 1993, he was a Research Fellow with the International Computer Science Institute, Berkeley, CA, USA. Since 1996, he has been a tenured Researcher with the FNRS, the Belgian National Funds for Scientific Research and the Co-Director of IRIDIA, the Artificial Intelligence Laboratory, ULB. He is the Inventor of the Ant Colony Optimization metaheuristic. His current research interests include swarm intelligence, swarm robotics, and metaheuristics for discrete optimization.

Prof. Dorigo was a recipient of the Italian Prize for Artificial Intelligence in 1996, the Marie Curie Excellence Award in 2003, the Dr. A. De Leeuw-Damry-Bourlart Award in Applied Sciences in 2005, the Cajastur International Prize for Soft Computing in 2007, an ERC Advanced Grant in 2010, the IEEE Frank Rosenblatt Award in 2015, and the IEEE Evolutionary Computation Pioneer Award in 2017. He is the Editor-in-Chief of *Swarm Intelligence*, and an Associate Editor or a member of the Editorial Boards of several journals on computational intelligence and adaptive systems. In 1993, he was a NATO-CNR Fellow. From 1994 to 1996, he was a Marie Curie Fellow. He is a Fellow of AAAI and EurAI.

Mary Katherine Heinrich received the B.Sc. degree from the University of Cincinnati, OH, USA, in 2013, the MAI degree from IAAC, Universitat Politécnica de Catalunya, Barcelona, Spain, in 2014, and the Ph.D. degree from the Centre for Information Technology and Architecture, Royal Danish Academy, Copenhagen, Denmark, in 2019.

From 2016 to 2018, she was a Recurring Visiting Ph.D. Researcher with the New England Complex Systems Institute, Cambridge, MA, USA, and from 2018 to 2019, a Research Associate with the Service Robotics Group, Institute of Computer Engineering, University of Lübeck, Luebeck, Germany. Since 2019, she has been a Postdoctoral Researcher with IRIDIA, the Artificial Intelligence Laboratory, Université Libre de Bruxelles, Brussels, Belgium. Her research interests include swarm intelligence, swarm robotics, and construction automation.