Self-reconfigurable Hierarchical Frameworks for Bearing-based Formation Control of Robot Swarms

Self-reconfigurable hierarchical frameworks for bearing-based formation control of robot swarms

Yuwei Zhang, Sinan Oğuz, Shaoping Wang, Emanuele Garone, Xingjian Wang, Marco Dorigo, 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 [1], 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 paper, we extend bearing rigidity to directed topologies and extend Henneberg constructions to generate hierarchical frameworks with bearing rigidity. We investigate three key self-reconfiguration problems: framework merging, robot departure, and framework splitting. We derive the mathematical conditions of these problems and then develop algorithms that preserve rigidity and hierarchy. Our approach can be used for formation control generally, as it can in principle 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 and report the results.

I. INTRODUCTION

In the last decades multi-robot systems have been proposed as the natural solution to carry out certain classes of missions, 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 multi-robot systems poses several problems, including limited scalability, a single point of failure in the coordinating agent, and potentially unrealistic communication infrastructure. To circumvent these problems, the swarm robotics community has successfully demonstrated that groups of robots can be controlled in a completely decentralized way [4]–[9]. However, as the size and speed of the swarm increases, the design and management of swarm-level behaviors becomes increasingly difficult.

To overcome this challenge, the idea of Mergeable Nervous Systems (MNS) [1], [10] has recently been introduced into the literature. 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], [10], [11].

The MNS paradigm is very promising for the control of large multi-robot systems—compared to strictly decentralized approaches—as it greatly simplifies the process of designing collective behaviors. In order for the MNS paradigm to extend to formation control of much larger robot 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. In this paper, we introduce a novel approach to self-reconfigurable hierarchical frameworks with mathematically provable properties. Our frameworks can be used for formation control in general, including for large robot swarms, by being coupled with a control law that makes use of bearing rigidity.

Rigidity graph theory is a fundamental mathematical tool to handle various problems in networked robotic systems, e.g. [12], [13]. 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 [14]–[20]. Bearing rigidity is a graph property that allows a formation to be maintained without external positioning, using only inter-agent measurements of bearing. Bearing (i.e., vector of arrival) can be sensed directly by onboard cameras or sensor arrays [16] in conjunction with an onboard inertial measurement unit. Bearing vectors remain unchanged during translational and scaling maneuvering of the formation [14],
enabling high flexibility in formation management. Bearing rigidity in $\mathbb{R}^2$ can be extended to an arbitrary dimensional space $\mathbb{R}^d$ ($d \geq 3$) [15]. Substantial progress on bearing-based formation control has been achieved in recent years (cf. [21]). However, the underlying graphs in existing approaches are assumed to be undirected. These are less natural than directed graphs when dealing with multi-robot 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 [22], it is heavily based on symmetry, and therefore cannot be simply applied to directed graphs [23]. Among the contributions of this paper, we address this research gap by presenting a solution for bearing rigidity under directed networks.

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 frameworks is the so-called Henneberg construction. Originally proposed for distance rigidity [24], Henneberg constructions have been extended to bearing rigidity in [23], [25]. Given the sequential nature of an Henneberg construction, it is not easy to implement it in a completely decentralized way, and therefore is not well suited for traditional swarm algorithms. However, we believe that this kind of construction is very suitable when designing hierarchical control structures for swarms. Based on this idea, in this paper we propose a novel approach to swarm control that combines the use of self-organized hierarchy (cf. MNS [1]) and Henneberg construction. Our approach has provable reconfiguration properties, and allows for formation control in swarms using behaviors that are simple to design. Self-reconfiguration is a crucial feature of this approach, and in this paper the following three reconfiguration scenarios will be explicitly analyzed: merging of frameworks, robot departure, and splitting of frameworks.

It is also important to investigate the preservation of rigidity in scenarios of framework self-reconfiguration. Attempts to solve the rigidity recovery problem can be found in the literature, in scenarios of merging [26], robot departure [27], [28] and splitting [29]. However, in order to add or remove the minimum edges to maintain rigidity, these existing solutions use global assessment and require centralized control. Approaches that depend on global assessment suffer from poor scalability, and therefore cannot be applied to large-scale robot swarms. We contribute a decentralized approach to self-reconfiguration with rigidity preservation, enabled by hierarchical control structures.

The main technical contributions of this paper can be summarized as follows:

1) We extend existing research on bearing rigidity, by addressing a directed rather than an undirected topology. We show that, in the case of directed graphs, bearing persistence is additionally required to evaluate bearing rigidity. We also provide necessary and sufficient conditions to uniquely determine a framework under a directed topology with asymmetric and lower triangular structure.

2) We propose a Hierarchical Henneberg Construction (HHC), to bridge the gaps between bearing rigidity and hierarchy. Bearing rigidity requires that each robot can refer to the bearings of at least two other robots. Therefore, our hierarchical framework has two leader robots, which occupy the first and second hierarchy layers. The hierarchical framework can be further decomposed into multiple minimal structures, each comprising two parent robots and one child robot. These minimal structures are much smaller in scale, and therefore more convenient to manage.

3) The mathematical conditions to preserve rigidity and hierarchy during framework merging, robot departure, and framework splitting are deduced, and guide the development of corresponding algorithms.

4) To demonstrate our proposed hierarchical frameworks and validate our theoretical results (including the preservation of hierarchy and bearing rigidity) in simulated experiments, we couple our approach with an example formation controller that makes use of bearing rigidity, and run experiments in four example scenarios of reactive formation control with moving leaders. On the basis of our hierarchical frameworks, we can flexibly manage the formation and reconfigure the robot swarm according to task requirements and environment constraints with respect to an arbitrary reference trajectory.

The remainder of this paper is organized as follows. In Sec. II, the foundational concepts of bearing rigidity and bearing persistence are presented. In Sec. III, we formulate three key problems addressed in this paper: framework construction, framework reconstruction, and validation of our frameworks using an example formation control law. These problems are addressed in Sec. IV, Sec. V and Sec. VI respectively. The conclusions are summarized in Sec. 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; $\mathbf{I}_d$ is the $d \times d$ identity matrix. $\mathbf{1}_n$ 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 $e_{ij} = (e_{ij}, e_{ij}) \in \mathcal{E}$, then the $i$-th robot can receive information from the $j$-th robot. $\mathcal{G}$ is undirected if $\forall e_{ij} \in \mathcal{E}$, there exists $e_{ij} \in \mathcal{E}$; otherwise, $\mathcal{G}$ is directed. We define the parent set of vertex $v_i$ as $\mathcal{P}_i = \{v_j \in \mathcal{V} | e_{ij} \in \mathcal{E}\}$, and the child set of $v_i$ as $\mathcal{C}_i = \{v_j \in \mathcal{V} | e_{ij} \in \mathcal{E}\}$.

It is assumed that for each edge $e_{ij} \in \mathcal{E}$, 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 $G$ associated with a configuration $p$, i.e., $(G, p)$. According to whether the underlying graph is directed or not, the framework is either an undirected framework or a directed framework.

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

### A. Bearing rigidity in undirected frameworks

To describe all the bearings in $(G, p)$, define the bearing function $F_B$ as $F_B(p) = \begin{bmatrix} g_1^T, g_2^T, \ldots, g_m^T \end{bmatrix}^T$, where $g_k$ corresponds to the $k$-th edge in graph $G$. Then, we can define the bearing rigidity matrix as

$$R_B(p) = \frac{\partial F_B}{\partial p} \in \mathbb{R}^{dn \times dn}. \quad (1)$$

**Definition 1 – Infinitesimal bearing rigidity [22]:** An undirected framework $(G, p)$ in $\mathbb{R}^d$ is infinitesimally bearing rigid (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 [22]:** An undirected framework $(G, p)$ in $\mathbb{R}^d$ is IBR if and only if $\text{rank}(R_B(p)) = dn - d - 1$, or equivalently $\text{Null}(R_B(p)) = \{ I_g \otimes I_d, p \}$.

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 non-infinitesimally bearing rigid frameworks** are presented in Fig. 1, where there clearly exist non-trivial 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 the framework. 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 non-trivial infinitesimal bearing motions.

\(^1\)Two kinds of trivial infinitesimal bearing motions exist: translational and scaling of the entire framework. More details are given in [22].

### B. Bearing persistence in directed frameworks

It is worth noting that rigidity is fundamentally an undirected notion, and therefore is not sufficient to characterize directed frameworks [30]. 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, Robot 2 and Robot 4 can be subsequently placed. However, Robot 2 and Robot 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, Robot 2, and Robot 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 paper 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 [22]

$$B_{ij} = \begin{cases} 0, & i \neq j, e_{ij} \notin E \\ -P_{g_{ij}}, & i \neq j, e_{ij} \in E \\ \sum_{g_{ik} \in P_i} P_{g_{ik}}, & i = j \end{cases} \quad (2)$$

where $B_{ij} \in \mathbb{R}^{dn \times dn}$ is the $ij$th block of a submatrix of $B$, and $P_{g_{ij}}$ is an orthogonal projection operator defined as $P_{g_{ij}} = I_d - g_{ij}g_{ij}^T$. 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 [31]:** $\text{rank}(B_{ii}) = d$ if and only if there exist at least two vertices $v_j, v_k \in P_i$, such that $g_{ij} \neq g_{ik}$.

We can now introduce the definition of Bearing persistence.

**Definition 2 – Bearing persistence [32]:** A directed framework $(G, p)$ in $\mathbb{R}^d$ is bearing persistent (BP) if $\text{Null}(B) = \text{Null}(R_B)$ [32]. 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 [22]. In this paper we focus on a special case where the bearing Laplacian is lower triangular, i.e.,

$$B = \begin{bmatrix} 0 & 0 \\ B_{2,1} & B_{2,2} \\ \vdots & \vdots \\ B_{n,1} & B_{n,2} & \cdots & B_{n,n} \end{bmatrix}.$$  \hspace{1cm} (3)

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

1. $(G, p)$ is IBR and BP.
2. $(G, p)$ can be uniquely determined up to a translational scaling factor.
3. Null $B = \text{Null } (RB) = \text{span } \{1_n \otimes I_d, p\}$.
4. rank $B = \text{rank } (RB) = \text{rank } (B_{2,2}) = d - 1$.
5. rank $B_{2,2} = d - 1$ and rank $B_{n,i} = d, \forall i \geq 3$.

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 Bearing Persistent (GBR-BP) graph: A directed graph $G$ is GBR-BP in $\mathbb{R}^d$ if there exists at least one configuration $p \in \mathbb{R}^{3n}$ such that $(G, p)$ in $\mathbb{R}^d$ is IBR and BP.

**Lemma 4:** Consider a directed graph $G = (V, E)$, with a lower triangular bearing Laplacian matrix $B$. $G$ is GBR-BP if and only if card $(P_2) = 1$ and card $(P_i) \geq 2, \forall i \geq 3$.

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 re-configuration strategies later. GBR-BP graphs have the following two properties.

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

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

**Lemma 5** and **Lemma 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 paper 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 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 [23], and is defined as follows.

**Hierarchical Henneberg construction (HHC).** 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 $v_1$ to $v_i$ in the directed graph $G$. 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 $v_3$ to the existing graph, incorporating two directed edges $e_{3,1}$ and $e_{3,2}$ to two existing vertices $v_1$ and $v_2$ in the graph. Then the hierarchy of vertex $v_3$ is defined as $h(v_3) = \max (h(v_1), h(v_2)) + 1$.
2) **Edge splitting:** Consider an existing vertex $v_3$ 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 $v_l$ together with three edges $e_{lk}$, $e_{jl}$, and $e_{li}$, where vertex $v_l$ is selected such that $h(v_l) \leq h(v_k)$. Then update the hierarchy of $v_k$ as $h(v_k) = \max(h(v_l), h(v_k)) + 1$ and the hierarchy of $v_l$ as $h(v_l) = \max(h(v_i), h(v_l)) + 1$.

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 $n_i$ 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

$$B = \begin{bmatrix} 0 & 0 & 0 \\ B_{2,1} & B_{2,2} & \cdots \\ \vdots & \vdots & \ddots \\ 0 & -B_{lj} & 0 & -B_{lk} & B_{1,1} & \cdots & \cdots & \cdots \\ \end{bmatrix}. \quad (4)$$

A graph $G = (V, E)$ generated via Henneberg construction is called a Laman graph [33]. It was proved in [31] that an undirected Laman graph is generically bearing rigid. Here, we further shows that a directed Laman graph is GBR-BP.

**Theorem 1.** A graph $G$, generated by HHC, is GBR-BP.

**Proof.** Following Lemma 4, $G$ is GBR-BP if and only if $\text{card} \ (P_2) = 1$ and $\text{card} \ (P_3) \geq 2, \forall i \geq 3$. Denote the graph consisting of $n$ vertices as $G_n = (V_n, E_n)$.

Firstly, we consider the case of $n = 2$. $G_2 = (V_2, E_2)$ is defined as $V_2 = \{v_1, v_2\}$ and $E_2 = \{e_{1,2}\}$. Note that the bearing Laplacian matrix of $G_2$ is lower triangular, and $\text{card} \ (P_2) = 1$. Therefore, the claim is true for $n = 2$.

Secondly, 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 $G_{n-1}$. According to HHC, there are the following two cases.

**Vertex addition:** Select two distinct vertices $v_j$ and $v_k$ from $G_{n-1}$. We add edges $e_{jk}$ and $e_{kn}$. It is trivial to verify that the bearing Laplacian matrix is still lower-triangular, as $j, k < n$.

Moreover, given that $G_{n-1}$ is GBR-BP and $\text{card} \ (P_n) = 2$, then $G_n$ is GBR-BP under Lemma 4.

**Edge splitting:** We select three vertices $v_k$, $v_l$, and $v_i$ from $G_{n-1}$ according to the requirements specified in the operation description. Then the new graph is given by $G_n = (V_n, E_n)$, where $V_n = V_{n-1} \cup \{v_n\}$ and $E' = E \setminus e_{jk} \cup \{e_{lj}, e_{ki}, e_{kn}\}$.

We relabel the vertices according to our labeling rule, such that the bearing Laplacian matrix is verified to be lower triangular.

**Algorithm 1** Constructing a hierarchical GBR-BP graph of $n$ ($n \geq 2$) robots in $\mathbb{R}^d$ ($d \geq 2$)

1: $i \leftarrow 0$;
2: Choose arbitrarily two robots from the swarm to define as leaders $v_1$ and $v_2$;
3: $G \leftarrow$ add vertices $v_1$ and $v_2$, and edge $e_{2,1} = (v_2, v_1)$;
4: $i \leftarrow i + 2$
5: $h(v_1) \leftarrow 0$, $h(v_2) \leftarrow 1$
6: While $i \leq n$ do
7: $i \leftarrow i + 1$
8: if Vertex addition is performed then
9: Choose arbitrarily two robots to define as $v_j$ and $v_k$ from $G$;
10: $G \leftarrow$ Add a vertex $v_l$ and two edges $e_{lj}$ and $e_{lk}$;
11: $h(v_l) \leftarrow \max(h(v_j), h(v_k)) + 1$;
12: else if Edge splitting is performed then
13: Choose arbitrarily one robot to define as $v_k$ from $G$, which has two parent robots $v_j$ and $v_l$;
14: Choose arbitrarily one robot to define as $v_l$ from $G$, satisfying $h(v_l) \leq h(v_k)$;
15: $G \leftarrow$ Remove edge $e_{lk}$, add one robot $v_l$ and three edges $e_{lj}$, $e_{kj}$ and $e_{kl}$;
16: $h(v_l) \leftarrow \max(h(v_j), h(v_l)) + 1$;
17: $h(v_k) \leftarrow \max(h(v_l), h(v_k)) + 1$;
18: end if
19: end while
20: return $G$.

We verify that $\text{card} \ (P_i) = 2$ is guaranteed $\forall 3 \leq i \leq n$. It follows from Lemma 3 that $G_n$ 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 Mergeable Nervous Systems 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 Sec. 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 Sec. VI.

**Remark 1:** Although the concept of hierarchy in the context of HHC was introduced in [23], our research differs and
contributes in two major ways. 1) Hierarchy was introduced as a concept in [23], but not fully investigated. We expand on the existing work and propose self-reconfiguration algorithms on the basis of hierarchical structures. 2) [23] handles frameworks geometrically. We expand on this by analyzing the rigidity of the hierarchical frameworks based on the notion of bearing persistence.

V. FRAMEWORK RECONSTRUCTION

In this section, we address the framework reconstruction problem, including merging frameworks, robot departure, and splitting frameworks.

A. Merging frameworks

A merging strategy for undirected frameworks that considers maintenance of bearing rigidity has been proposed in [26]. We build upon [26] by extending to the case of directed graphs and maintenance of the hierarchical structure and bearing persistence.

To start, consider two directed frameworks: $(G_a, p_a)$ with $n_a$ robots, and $(G_b, p_b)$ with $n_b$ robots. 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 $G_a = (V_a, E_a)$ and $G_b = (V_b, E_b)$, generated by HHC. Denote two leaders of framework $(G_a, p_a)$ as $v_{a1}$ and $v_{a2}$ and perform the following sequence of operations: 1) Select two vertices $v_{a1}, v_{a2} \in V_a$; 2) Add three edges $e_{a1} = (v_{a1}, v_{a2}), e_{a2} = (v_{a1}, v_{a3}),$ and $e_{a3} = (v_{a2}, v_{a3})$. The resulting post-merged graph $G = (V, E)$ defined by $V = V_a \cup V_b$ and $E = E_a \cup E_b \cup \{e_{a1}, e_{a2}, e_{a3}\}$ is GBR-BP.

**Proof.** We add two edges to $v_{a1}$ and one edge to $v_{a2}$, which results in $\text{card}(P_i) = 2$ 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 $(G_a, p_a)$ should be updated as $h(v_{a1}) \leftarrow h(v_{a1}) + \max \{h(v_{a2}), h(v_{a3})\} + 1$. 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 can be calculated from this algorithm that the time complexity for merging is $O(m)$. It is worth noting that the merging process is allowed to be parallel. Therefore, in practice, the time complexity will often be lower than $O(m)$. 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. Therefore, we can achieve a faster self-organization process via parallelization.

**Algorithm 2** Merging $m$ GBR-BP graphs $G_1, G_2, \ldots, G_m$ into one GBR-BP graph $\bar{G}$ in $\mathbb{R}^d (d \geq 2)$

1. $\bar{G} \leftarrow G_1$;
2. for $k = 2 \rightarrow m$ do
3. Select two vertices $v_i$ and $v_j$ from $\bar{G}$;
4. Select leader vertices $v_{i1}$ and $v_{i2}$ from $G_k$;
5. Add edges $e_{a1} = (v_i, v_{i1}), e_{a2} = (v_i, v_{i2}),$ and $e_{a3} = (v_j, v_{i2})$;
6. $\bar{G} = \bar{G} \cup \{\bar{V}, \bar{E}\}$, where $\bar{V} = V \cup V_k$ and $\bar{E} = E \cup E_k \cup \{e_{a1}, e_{a2}, e_{a3}\}$;
7. Update the hierarchy of vertices in $G_k$ as $h(v_{i2}) \leftarrow h(v_{i2}) + \max \{h(v_i), h(v_j)\} + 1$;
8. end for
9. return $\bar{G}$;

B. Robot departure

In this subsection, 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: outer node (i.e., no child) and 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 hierarchical framework $(G, p)$ with $n$ robots. We assume that the vertex $v_{in}$ is an outer node and its parent vertices are labelled as $v_i$ and $v_j$.

**Theorem 3:** Given a GBR-BP graph $G = (V, E)$ generated by HHC, remove an outer node vertex $v_{in}$ and two associated edges $e_{in}$ and $e_{jn}$, the graph $G' = (V', E')$ defined by $V' = V \setminus \{v_{in}\}$ and $E' = E \setminus \{e_{in}, e_{jn}\}$ 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 $G = (V, E)$ generated by HHC, if an inner vertex $v_i$ leaves the framework, let its position in the framework (including hierarchy and connected edges) be inherited by the one of its children vertices $v_{in}$, which has the highest hierarchy in $C_i$. In other words, $\forall v_j \in C_i, h(v_{in}) \geq h(v_j)$. If $v_{in}$ is an outer node, Theorem 3 yields that the
resultant graph after performing the inheriting operation is GBR-BP. If \( v_{in} \) is an inner node, we can continue performing the inheriting 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 inheriting operation. The time complexity of our proposed robot departure algorithm can be calculated as \( O(n) \). In contrast to our approach, existing methods such as [28] require an optimal repairing solution from the global perspective to find the necessary edges to maintain the rigidity.

### C. Splitting frameworks

In this subsection, we consider the case where a framework with at least four robots is split into several disjoint sub-frameworks, each consisting of at least two robots. Similar to the merging operation, the main difficulty of the splitting operation is preservation of the bearing rigidity, persistence, and hierarchy of the sub-frameworks 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 sub-frameworks.

We use a special graph called Z-link, originally proposed in [34] and employed in [29] for undirected graphs. We extend this existing research to directed Z-links. We denote Z-link by \( Z = (V_Z, E_Z) \), where \( |V_Z| = 4 \) and \( |E_Z| = 3 \), as shown in Fig. 7. The following definition determines the existence of a Z-link in a graph \( G \).

**Definition 4 – Z-link:** Consider a directed graph \( G = (V, E) \). Two disjoint subgraphs \( G_a = (V_a, E_a) \) and \( G_b = (V_b, E_b) \) are said to be connected via a Z-link if the following two conditions hold.

1) There exist four distinct vertices \( v_{a1}, v_{a2} \in V_a \) and \( v_{b1}, v_{b2} \in V_b \), such that the graph among these four vertices is a Z-link.

2) \( V_a \cup V_b = V, V_a \cap V_b = \emptyset, E_a \cap E_b = \emptyset, \) and \( E = E_a \cup E_b \).

**Theorem 4:** Given a GBR-BP graph \( G = (V, E) \), let \( G_a = (V_a, E_a) \) and \( G_b = (V_b, E_b) \) be two disjoint subgraphs of \( G \), which are connected via a Z-link. Then, \( G_a \cup G_b \) is GBR-BP.

**Proof:** We only show that \( G_a \) is GBR-BP \( \Rightarrow \) \( G_b \) is GBR-BP, because the reverse is the same.

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

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

We augment \( G_a \) to \( G_a^+ = (V_a^+, E_a^+, \{e\}) \), defined by \( V_a^+ = V_a \cup \{v_{a1}, v_{a2}\} \) and \( E_a^+ = E_a \cup E \cup \{e\} \). Denote \( B_a^+ \) as the bearing Laplacian matrix of \( (G_a^+, p_a) \), where \( p_a = [p_a^T, p_{n_a+1}^T, p_{n_a+2}^T]^T \in \mathbb{R}^{dn_a+2} \).

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

\[
B^+ = \begin{bmatrix}
B_a^+ & 0^T \\
0 & 0^T
\end{bmatrix}
\begin{bmatrix}
0^T & 0^T
0^T & B_b
\end{bmatrix}.
\]

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

Let \( q = [p_1^T, \ldots, p_{n_a+1}^T, p_{n_a+2}^T, q_{n_a+3}^T, \ldots, q_n^T]^T \). Eq. (5) yields \( B^+ q = 0 \). Note that \( q \notin \text{span} \{1_n, 1_d, p\} \), i.e., \( G^+ \) is not GBR-BP, which is a contradiction. Therefore, \( G_b \) 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 \( G_a \) and \( G_b \), 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 firstly find a GBR-BP
subgraph $G_a$, and secondly construct a $Z$-link between $G_a$ and $G_b$. After the removal of the $Z$-link edges, we obtain two GBR-BP subgraphs. We now present our 2-step algorithm to split the framework, exploiting the triangularity in Eq. (4).

**Step 1: Find a GBR-BP subgraph** $G_a$.

Given a framework generated by HHC with $n$ robots, the Bearing Laplacian submatrix of the first $n_a$ robots always satisfies a triangular structure (cf. the triangularity in Eq. (4)). Therefore, we can verify $G_a$ as GBR-BP according to the first $n_a$ robots, as stated in the following Theorem.

**Lemma 7:** Given a GBR-BP graph $G = (V, E)$ generated by HHC, let $G_a = (V_a, E_a)$ represent a subgraph describing the interactions corresponding to first $n_a$ vertices, i.e., $V_a = \{v_1, \ldots, v_{n_a}\}$. Then, $G_a$ is GBR-BP.

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

$$B = \begin{bmatrix} B_{a} & 0 \\ B_{b1} & B_{b2} \end{bmatrix},$$

where $B_a \in \mathbb{R}^{d n_a \times d n_a}$ 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 $G_a$ is GBR-BP.

**Step 2: Construct a $Z$-link between two subgraphs.**

Let $G_b = (V_b, E_b)$ represent the interactions among the remaining vertices, i.e., $V_b = \{v_{n_a+1}, \ldots, v_n\}$. Corresponding to Definition 4, $Z$-link construction comprises the following two parts.

1. Let $P_{n_a+1} = \{v_{p1}, v_{p2}\}$. Remove the outgoing edges of $v_{n_a+2}$, then add edges $e_{p1(n_a+2)}$ and $e_{p2(n_a+2)}$.
2. For all $v_i \in V_b \setminus \{v_{n_a+1}, v_{n_a+2}\}$, if its parent $v_j \in V_a$, then remove $e_{j,i}$. To preserve rigidity, add new edge $e_{ki}$, where $v_k \neq v_i$ 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 4. It can be calculated from this algorithm that the time complexity for splitting is $O(mn)$.

**Algorithm 4** Splitting one GBR-BP graph $G$ into $m$ GBR-BP graphs: $G_1, G_2, \ldots, G_m$ in $\mathbb{R}^d$ ($d \geq 2$).

1. $n_0 = 0$;
2. for $k = 1 \rightarrow m$ do
3. $V_k \leftarrow \{v_{n_k-1+1}, \ldots, v_{n_k-1+n_k}\};$
4. Select vertex $v_{n_k-1+1}$ as the first leader of $G_k$, and denote its parents as $v_{p1}$ and $v_{p2}$;
5. Select vertex $v_{n_k-1+2}$ as the second leader of $G_k$;
6. $G_k \leftarrow$ Remove the connections between $v_{n_k-1+2}$ and its parents, and add edges $e_{ki} = (v_{n_k-1+1}, v_{n_k-1+2})$ and $e_{k2} = (v_{p1}, v_{n_k-1+2});$
7. $h(v_{n_k-1+2}) \leftarrow h(v_{n_k-1+1}) + 1$;
8. $h(v_i) \leftarrow h(v_i) + 2, \forall i \in \{n_k, n_k+1, \ldots, n_k+n_k\};$
9. for $i = n_k-1 + 3$ to $n_k-1 + n_k$ do
10. for $v_j \in P_i$ do
11. if $v_j \notin V_k$ then
12. $G_k \leftarrow$ remove edge $e_{ji}$;
13. $G_k \leftarrow$ add new edge $e_{ji}$, where $v_j \in V_k$ is chosen such that $v_j \notin P_i$ and $h(v_j) < h(v_i)$;
14. end if
15. end for
16. Update $h(v_i)$ according to the hierarchy of its parents;
17. end for
18. $E_k \leftarrow \{e_{ij} \subset \mathcal{E}|v_i, v_j \in V_k\};$
19. end for
20. $G_k \leftarrow$ Remove edges in constructed $Z$-links;
21. return $G_1, G_2, \ldots, G_m$.

In practice, the time complexity will often be lower than $O(mn)$.

A splitting example is presented in Fig. 8. Note that Algorithm 4 can split the framework into at least two subgraphs with arbitrary size no less than 2, which provides more flexibility in managing the size of the framework.

VI. VALIDATION WITH AN 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 a 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 4 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 $\dot{p}_i = v_i$, where $p_i \in \mathbb{R}^d$ is the inertial position of $i$th robot and $v_i \in \mathbb{R}^d$ 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 [20].)

A. Example scenario 1: Achieving the desired formation

Consider a swarm with $n$ robots, with topology characterized by a directed graph $G = (V, E)$ 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., $\{g_{ij} \, | \, e_{ij} \in E\}$. We assume that the positions of the neighboring robots do not coincide, i.e., $\forall e_{ij} \in E$ 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 $\{g_{ij} \, | \, e_{ij} \in E\}$ among the robots, 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 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 $\{g_{ij} \, | \, e_{ij} \in E\}$ is characterized by Lemma 1 in [23].

According to Lemma 1 in [23], 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 paper 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 $p_2(t) = p_1(t) - d_{21}^*(t)g_{21}$, at all time $t > 0$. 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

$$\dot{p}_i = u_i = -c \left(P_y g_{ij}^* + P_y g_{ik}^* \right) + \hat{p}_i^*, \quad (7)$$

where $c$ is a positive constant to be tuned and $\hat{p}_i^*$ is a feedforward term given below.

$$\hat{p}_i^*(t) = \left(P_{y_{ij}} + P_{y_{ik}} \right)^{-1} \left(P_{y_{ij}} \dot{p}_j^*(t) + P_{y_{ik}} \dot{p}_k^*(t) \right), \quad (8)$$

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

Remark 2: The control law (7) is inspired by [15], 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 paper. 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.

By employing a similar stability analysis as shown in Theorem 1 of [23], we can also demonstrate that the formation tracking error $e_i(t) = p_i(t) - p^*_i(t)$ 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 3: The only parameter to be tuned in Eq. (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 trade-off 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 hierarchical framework shown in Fig. 4(a), and the target formation is a cube. The motion of leader $v_1$ and time-varying distance $d_{21}^*$ are shown in Eq. (9). The controller parameter is chosen as $c = 5$. Fig. 9(a) depicts the trajectory of eight robots. 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
\[
\begin{align*}
p_1(t) &= [0.3t, 40 \sin(\pi t/200), 40 \sin(\pi t/200)]^T, d^*_{21}(t) = 20 - 10 \sin(\pi t/200), t \leq 100 \\
p_1(t) &= [0.3t, 40, 40]^T, d^*_{21}(t) = 10, 100 \leq t \leq 300 \\
p_1(t) &= [0.3t, 40 \sin(\pi(t - 200)/200), 40 \sin(\pi(t - 200)/200)]^T, d^*_{21}(t) = 20 - 10 \sin(\pi(t - 200)/200), t \geq 300
\end{align*}
\]
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 sub-frameworks will result. Thus, from $100$ s to $200$ s, robots $v_3$ and $v_6$ will be the leaders of Swarm B and the two sub-frameworks will move separately. The simulation results show that two sub-frameworks satisfy the bearing rigidity, because the formations are maintained after splitting, thus validating Theorem 4 and Lemma 7 of our approach.

VII. CONCLUSIONS

This paper investigates the construction of self-reconfigurable hierarchical frameworks for formation control of robot swarms, based on bearing rigidity under directed topologies. Self-organized hierarchical control is a promising approach to ease the design and management of collective behaviors in robot swarms. Hierarchical frameworks have already been demonstrated in practical studies using the Mergeable Nervous Systems paradigm [1], [10]. If strong theoretical foundations are also developed, then self-organized hierarchy can greatly simplify the design of collective behaviors in large-scale swarms of fast robots. This paper represents 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 introduce a hierarchy property into conventional Henneberg construction and extend bearing rigidity to directed graphs. We study self-reconfigurable hierarchical frameworks in three key framework reconstruction problems: merging, robot departure, and splitting. Finally, we demonstrate our frameworks by combining them with an example formation controller, and validate our theoretical results of hierarchy and rigidity preservation during reconfiguration via simulation experiments in four example scenarios.

In future research we intend to verify our approach using real ground robots and drones. We will also use our hierarchical frameworks to develop a unified maintenance strategy to rigidify non-rigid frameworks. In addition, we will extend our formation controller to address practical issues such as obstacle avoidance.

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 $\text{null} (B) = \text{span} \{ 1_n \otimes I_d, p \}$, it is equivalent to show $\forall q = [q_1, \ldots, q_n]^T \in \text{null} (B)$, $q = ap + 1_n \otimes b$, where $a \in \mathbb{R} \setminus \{0\}$ and $b \in \mathbb{R}^d$.

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

For robot 2, $q_2$ satisfies $B_{2,1} (q_2 - q_1) = 0$. If $B_{2,1} = 0$, 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 $1 \leq i \leq n$, 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

$$
(B_{3,1} + B_{3,2}) q_3 = (B_{3,1} + B_{3,2}) q_1 + \alpha B_{3,2} (p_2 - p_1) = (B_{3,1} + B_{3,2}) q_1 + \alpha (p_3 - p_1),
$$

where the last equality uses $B_{3,1} (p_3 - p_1) + B_{3,2} (p_3 - p_2) = 0$. 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 non-infinitesimal 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 $1 \leq k \leq i - 1$. For robot $i$, we have

$$
\sum_{j=1}^{i-1} B_{ij} q_j = \sum_{j=1}^{i-1} B_{ij} q_j = \sum_{j=1}^{i-1} B_{ij} q_j = \sum_{j=1}^{i-1} B_{ij} (q_j + \alpha (p_j - p_1)),
$$

where the last equality uses $\sum_{j=1}^{i-1} B_{ij} (p_j - p_1) = 0$. Via similar analysis for robot 3, the uniqueness of the framework ensures that $\sum_{j=1}^{i-1} B_{ij}$ is non singular, 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 $q = \alpha p + 1_n \otimes (q_1 - \alpha p_1)$. This implies that $\forall q \in \text{null} (B)$, $q \in \text{span} \{ 1_n \otimes I_d, p \}$.

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

$$
Bq = \begin{bmatrix}
\sum_{j=1}^{n-1} B_{ij} (q_i - q_j) \\
\vdots \\
\end{bmatrix} = 0,
$$

then $q \in \text{null} (B) = \text{span} \{ 1_n \otimes I_d, p \}$.

(4) $\Rightarrow$ (5): According to Eq. (2), $\text{rank} (B_{2,2}) = d - 1$ if $e_{i,j} \in E$, and $\text{rank} (B_{3,2}) = 0$ otherwise. Assume $e_{i,j} \notin E$, then $p_1$ and $p_2$ can be placed arbitrarily, which is a contradiction with statement (2). For robot $i$ ($i \geq 3$), 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 \( \text{card}(\Pi_i) = 0 \), or randomly placed along a line, if either card(\( \Pi_i \)) = 1 or \( v_{ij}, v_{k} \in \Pi_i, g_{ik} \neq g_{kj} \). In each of these cases, the framework cannot be unique. Therefore, \( \text{rank}(B_{ij}) = d \).

\[
\sum_{n}^i \text{rank}(B_{ij}) = d - n - 1.
\]

Note that \( \text{rank}(B) \leq d - n - 1 \) exists, hence rank \( (B) = d - n - 1 \).

### APPENDIX B

**Proof of Lemma 4**

**Proof:** Necessity: If \( G \) is GBR-BP, there exists a configuration \( p \) such that \( (G, p) \) is IBR and BP. Therefore, statement (5) in Lemma 3 should be satisfied. Since rank \( (B_{2,2}) = d - 1 \), robot 1 should be the parent of robot 2. For \( i > 3 \), rank \( (B_{ij}) = d \) if and only if at least two of \( \{g_{ik}\}_{k \neq i} \) are not collinear. Thus, card \( (\Pi_i) \geq 2 \). Sufficiency: If card \( (\Pi_2) \geq 1 \) and card \( (\Pi_i) \geq 2 \), we should find a configuration \( p = \{p_1, p_2, \ldots, p_n\} \in \mathbb{R}^d \), such that \( (G, p) \) 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 rank \( (B_{2,2}) = d - 1 \). For \( p_i (i > 3) \), because the position of its parents have been determined, \( p_i \) can be selected such that there exist at least two vertices \( v_{ij}, v_{k} \in \Pi_i \), with \( g_{ik} \neq g_{kj} \), which guarantees rank \( (B_{ij}) = d \). In this way, we find one configuration \( p \), such that statement 5 in Lemma 3 is satisfied. Thus, \( G \) is GBR-BP.

### REFERENCES