

Persistent Multi-Robot Formations with Redundancy

Alyxander Burns, Bernd Schulze, and Audrey St. John

Abstract A multi-robot formation composed of autonomous agents may need to maintain an overall rigid shape for tasks such as collective transport of an object. To distribute control, we construct leader-follow formations in the plane that are *persistent*: designated “leader” robots control the movement of the entire formation, while the remaining “follower” robots maintain directed local links sensing data to other robots in such a way that the entire formation retains its overall shape. In this paper, we present an approach based on rigidity theory for constructing persistent leader-follower formations with redundancy; specified robots may experience sensor link failure without losing the persistence of the formation. Within this model, we consider the impact of special positions due to certain geometric conditions and provide simulation results confirming the expected behavior.

1 Introduction

For applications such as collective transport, multi-robot formations need to maintain a global shape. To do so in a distributed fashion, we focus on formations composed of autonomous agents that use local sensing to maintain a global rigid structure. In particular, we consider *persistent leader-follower* formations where designated leader robots control the trajectory of the entire formation; the remaining robots autonomously sense and adjust their positions locally to follow specified robots in a way that maintains the global structure.

Alyxander Burns, Audrey St. John (e-mail: {burns221, astjohn}@mholyoke.edu)
Computer Science Department, Mount Holyoke College, South Hadley, MA
Partially supported by NSF IIS-1253146 and the Clare Boothe Luce Foundation

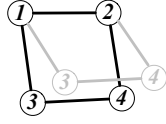
Bernd Schulze (e-mail: b.schulze@lancaster.ac.uk)
Department of Mathematics and Statistics, Fylde College Lancaster University, Lancaster, UK
Supported by EPSRC grant EP/M013642/1

In our model, each autonomous robot is represented as a point in the plane, and we work with range-only measurements, represented as distance constraints between pairs of points. This model is known in an area called “rigidity theory” as the 2D *bar-and-joint framework* (see, e.g., [19]) and is well-understood, with a quadratic algorithm for determining the bar-and-joint rigidity properties [8]. While rigidity theory has been applied to the construction and analysis of formations of autonomous agents [11, 3], the approach assumes undirected constraints, leading to a model where both agents would be responsible for the constraint. Since this may increase sensing and communication costs, a “persistence theory” for directed distance constraints between points was proposed by Hendrickx et al. [5] (see also [4]), effectively cutting costs in half by assigning one of the two agents to be responsible for sensing and maintaining a distance. Unlike decentralized approaches for collective transport where robots maintain constraints to the transported object (e.g., [7, 16]), a persistent formation could be used to carry delicate items, such as a partially constructed vehicle. In particular, we use a leader-follower architecture [2]; as described in [15], local sensing and communication can achieve specific geometric formations, allowing dynamic adaptation based on the surrounding environment.

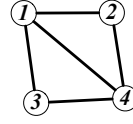
Contributions. In this paper, we focus on accommodating sensing and communication failures by incorporating redundancy into our model. Redundancy is well-understood in (undirected) rigidity theory, and the associated objects form the foundation for the main contribution of this paper: an approach for constructing (directed) persistent leader-follower formations with redundancy.

We work within the basic model of persistence, following the definitions from [5], and present a class of directed graphs where any edge from a vertex with out-degree 3 is redundant; after removal of such an edge, the resulting formation remains persistent. Algorithms for constructing these graphs, as well as simulation results confirming the expected behavior of acyclic formations, are presented. We also include a discussion of the impact of special geometric conditions that can affect the “generic” behavior of the combinatorial model. To the best of our knowledge, these graphs are the first to incorporate redundancy into persistent formations. While the redundancy is restricted to specified sets of edges, it is a first step towards the stronger notion of *redundantly persistent* formations, defined in Section 3, where any edge in the formation could be removed.

Structure. In Section 2, we provide an overview of the relevant definitions and results from rigidity and persistence theory. We present an approach for constructing persistent leader-follower formations with redundancy in Section 3 before considering special geometric conditions that may affect persistence in Section 4. However, restricting to acyclic formations implies that such special conditions do not impact our construction, and we present simulation results (Section 5) verifying our approach. We conclude with future directions in Section 6.



(a) A flexible framework with another (gray) incongruent embedding satisfying the distance function.



(b) Adding the edge 14 results in a minimally rigid framework.

Fig. 1 Flexible and rigid frameworks in the plane.

2 Preliminaries

Let $G = (V, E)$ be an undirected graph with vertex set $V = [1..n]$ and edge set E of unordered pairs of vertices. An *embedding* of G in the Euclidean plane is an assignment $\mathbf{p} \in (\mathbb{R}^2)^n$ of the vertices to points in the plane; the pair (G, \mathbf{p}) is called a *framework*. Another embedding \mathbf{q} is *congruent* to \mathbf{p} if $\|\mathbf{p}_i - \mathbf{p}_j\| = \|\mathbf{q}_i - \mathbf{q}_j\|$ for every pair of vertices i and j . Given a framework, we can extract a distance function $d : E \rightarrow \mathbb{R}$, where $d(ij) = \|\mathbf{p}_i - \mathbf{p}_j\|$. If all \mathbf{q} in the neighborhood of \mathbf{p} satisfying the distance function d are congruent to \mathbf{p} , the framework is *rigid* and *flexible* otherwise; refer to Figure 1. The rigid framework of Figure 1(b) is *minimally rigid* as the removal of any edge results in a flexible framework.

For a given graph, almost all associated embeddings, called *generic* embeddings, share the same rigidity properties (see, e.g., [19]). Therefore, we may call a graph *generically* rigid or flexible, referring to the behavior of generic embeddings. The formal definition of genericity is captured by a polynomial whose vanishing indicates a non-generic embedding, or *special position*, and is outside the scope of this paper. The impact of special positions is discussed in Section 4.

For multi-robot formations, where minimizing the cost of communication and sensing is desirable, we work with a notion closely related to rigidity called *persistence*. We build upon the foundations of [5] and include here only the relevant definitions and results. Persistence is framed in terms of a directed graph and intuitively defines the directed analog of rigidity. One can interpret each of the vertices as representing an autonomous agent with out-going edges specifying distance constraints to neighbors that it is responsible for satisfying. This eliminates the cost for sensing and communication costs on one endpoint of a constraint edge, which would be required if working with undirected graphs and rigidity. If (1) every agent can find a position to satisfy its distance constraints, and (2) the corresponding framework is rigid, then the formation is considered persistent.

Let $H = (V, E)$ be a directed graph with vertex set $V = [1..n]$ and edge set E of ordered pairs of vertices. For clarity, we denote a directed edge from the source i to j with \overrightarrow{ij} to contrast with an undirected edge ij . For a graph H and an embedding \mathbf{p} , the pair (H, \mathbf{p}) is called a *formation*. Given a formation, we can extract the distance function d as before.

We can now state the technical definitions from [5] for persistence. Given d , let $\mathbf{q} \in (\mathbb{R}^2)^n$ be an embedding of the vertices of H . If $\vec{ij} \in E$ and $\|\mathbf{q}_i - \mathbf{q}_j\| = d(\vec{ij})$, then the edge \vec{ij} is *active in* \mathbf{q} . The set $A_{\mathbf{q}}(i)$ denotes the set of active edges in \mathbf{q} whose source is i . The position \mathbf{q}_i is *fitting* for vertex i if there does not exist another embedding \mathbf{q}^* such that (1) $\mathbf{q}_i^* \neq \mathbf{q}_i$, (2) $\mathbf{q}_j^* = \mathbf{q}_j$ for all $j \neq i$, and (3) $A_{\mathbf{q}}(i) \subsetneq A_{\mathbf{q}^*}(i)$. Intuitively, a position is fitting for a vertex if it cannot be moved to satisfy additional constraints. If the positions for all vertices are fitting, then \mathbf{q} is a *fitting embedding*. With this notion of fitting embeddings formalized, we can define persistence.

Definition 1. Let (H, \mathbf{p}) be a formation. If all fitting embeddings \mathbf{q} in the neighborhood of \mathbf{p} are congruent to \mathbf{p} , then the formation is *persistent*.

See Figure 2 for examples of persistent and non-persistent formations. While the formations in Figures 2(a) and 2(b) have the same underlying (rigid) undirected graph, only one is persistent. The embedding shown in Figure 2(c) is fitting for the formation in Figure 2(b) as each vertex is in a position that maximizes the number of its out-going constraints; vertex 4 cannot satisfy the dashed constraint to 3 without violating at least one of its other constraints to 1 or 2. Since it is not congruent, this certifies that the formation in Figure 2(b) is not persistent.

As with rigidity, almost all embeddings of a directed graph exhibit the same persistence properties, so we may refer to a directed graph as *generically persistent*. We rely on the following result of [5]:

Theorem 1 (Theorem 3 of [5]). *A graph is generically persistent if and only if the underlying undirected graph of every subgraph obtained by removing out-edges from vertices with out-degree > 2 until all vertices have out-degree ≤ 2 is generically rigid.*

Leader-follower formations. In this paper, we consider a specific type of persistent formations called *leader-follower* formations. In the persistent formations of Figures 2(a) and 2(d): only vertex 1 has out-degree 0 and is called the *leader*, only vertex 2 incident to it has out-degree 1 and is called the *co-leader*, and all other vertices (3 and 4) have out-degree ≥ 2 and are called *followers*. Since a point in 2D has two degrees of freedom (translation along the x- and y-axes), but an entire formation has three (translation along the x- and y-axes along with rotation about the origin), the simplest leader-follower formation must have both a leader and a co-leader; the entire formation cannot be controlled by a single lead point agent.

3 Redundancy for persistence theory

Communication links and sensors can fail, motivating the need for redundancy in a multi-robot system. In this section, we present an approach for constructing persistent leader-follower formations with redundancy.

We begin by reviewing redundancy in rigidity; a graph is generically *redundantly rigid* if removing any edge results in a rigid graph. Minimality is defined as follows:

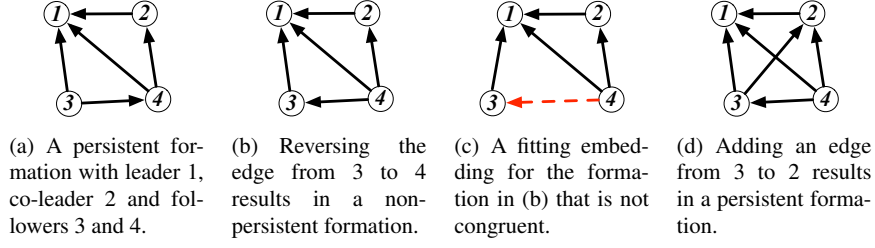


Fig. 2 Persistent and non-persistent frameworks in the plane.

an undirected graph is a generic *rigidity circuit* if removing any edge results in a generically minimally rigid graph. Circuits are standard in matroid theory (see, e.g., [12]) and the *2D bar-and-joint rigidity matroid* captures the behavior of the distance constraints described in Section 2 [18]. In this section, we only consider the generic behavior of graphs; for brevity, we omit the word “generically” for the remainder. The smallest example of a rigidity circuit can be seen in Figure 2(d); removing any edge from the (undirected) K_4 graph gives a minimally rigid graph.

We analogously define a directed graph to be *redundantly persistent* if the removal of any edge results in a persistent graph. However, the behavior of redundant rigidity does not easily extend to the persistence model. Refer back to the formation in Figure 2(d). While its underlying undirected graph is redundantly rigid, the formation is not redundantly persistent; without the edge $\overrightarrow{32}$, the resulting formation (of Figure 2(b)) is no longer persistent. We leave the question of redundantly persistent graphs open; it is challenging to even come up with a simple formation that satisfies the definition.

In the remainder of this section, we present a class of graphs with a more restricted notion of redundancy. These arise from considering leader-follower formations whose underlying undirected graphs are rigidity circuits, beginning with the following result.

Proposition 1. *Let G be a rigidity circuit. Then there exists a leader-follower orientation of G that is persistent.*

Proof. Let $G = (V, E)$ be a rigidity circuit and let $e = ij \in E$ be any edge. Then $G' = (V, E \setminus \{e\})$ is a minimally rigid graph. By using an algorithm called the (2, 3)-pebble game [8, 10], there exists an orientation of G' where every vertex has out-degree at most 2. Furthermore, we can use “pebble collection” moves in the pebble game to find an orientation H where exactly one vertex v_L has out-degree 0, another vertex v_C incident to v_L , has out-degree 1 and all other vertices have out-degree 2. By Theorem 1, this formation is persistent; there are no vertices with out-degree > 2 , so we only need to consider the underlying undirected graph G' of H , which is (minimally) rigid.

We now add the edge ij back to the formation, orienting it as \overrightarrow{ij} if $i \neq v_C, v_L$ and \overrightarrow{ji} otherwise. By this construction, the source of the edge e now has out-degree 3.

Consider the undirected graphs underlying the three subgraphs obtained by dropping each out-edge from the source of e ; since G is a rigidity circuit, each is (minimally) rigid. Therefore, by Theorem 1, this formation is persistent. \square

Note that the proof is constructive, as captured by Algorithm 1.

Algorithm 1 Constructing a persistent leader-follower formation from a rigidity circuit.

Input: a rigidity circuit $G = (V, E)$, a desired leader vertex v_L and a desired co-leader vertex v_C incident to v_L .

Output: a persistent leader-follower formation.

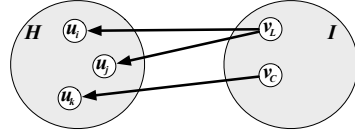
1. Remove any edge $e = ij \neq v_C v_L \in E$.
 2. Play the (2,3)-pebble game on $G' = (V, E \setminus \{e\})$ to obtain a directed graph H .
 3. Use pebble collection moves on H to collect 2 pebbles on v_L and one pebble on v_C .
 4. Output the resulting directed graph with the additional edge \vec{ij} , if $i \neq v_C, v_L$, or \vec{ji} otherwise.
-

Algorithm 1 runs in $O(n^2)$ time; Steps 1 and 4 are constant, and Steps 2 and 3 take $O(n^2)$ and $O(n)$ time, respectively [10]. The produced persistent formation has the following properties: (1) there is a single leader vertex v_L ; (2) there is a single co-leader vertex v_C incident to the leader; (3) there is exactly one follower vertex (the source of e) that has out-degree 3 and removal of any of its out-edges will maintain persistency; and (4) every other follower vertex has out-degree 2. As an example, refer again to Figure 2(d). It was constructed by first dropping the edge 24 . Then the pebble game algorithm was executed, giving the orientation without $\vec{42}$, with vertex 1 as the leader and 2 the co-leader. Adding the edge back in with direction $\vec{42}$ (since 2 is the co-leader) gives a persistent leader-follower formation, where vertex 4 is the one follower vertex with out-degree 3.

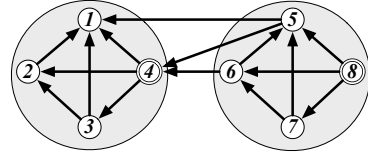
The formations produced by Algorithm 1 contain redundancy via the out-going edge set of the vertex with out-degree 3. For the K_4 example, this implies that, if a sensor link were to fail, there is a 50% chance of that failure not impacting the persistence of the formation; exactly 3 of the 6 edges are in the redundant set of edges leaving vertex 4. While this does not give a formation where the redundancy is uniformly distributed through the formation, it is a step towards a more robust theoretical model.

Recursively constructing persistent formations. By using the formations from Algorithm 1 as *seed* formations, we can recursively construct larger formations with more vertices permitting sensor failures. We rely on the following result from [6], specialized to our setting. See Figure 3(a) for a visual depiction of the construction.

Proposition 2 (Proposition 3 of [6]). *Let $H = (U, E)$ and $I = (V, F)$ be persistent leader-follower graphs with leaders $u_L \in U$, $v_L \in V$ and co-leaders $u_C \in U$ and $v_C \in V$. Then the graph $(U \cup V, E \cup F \cup \{e = \vec{v_L u_i}, f = \vec{v_L u_j}, g = \vec{v_C u_k}\})$ is a persistent leader-follower graph, where $u_i, u_j, u_k \in U$ and $|\{u_i, u_j, u_k\}| > 1$.*



(a) The construction adds 3 out-edges from the leader and co-leader of one formation to the other.



(b) Applying the construction to two persistent leader-follower “seeds” formations.

Fig. 3 Proposition 2’s recursive construction allows additional (double-outlined) vertices with out-edge sets containing redundancy.

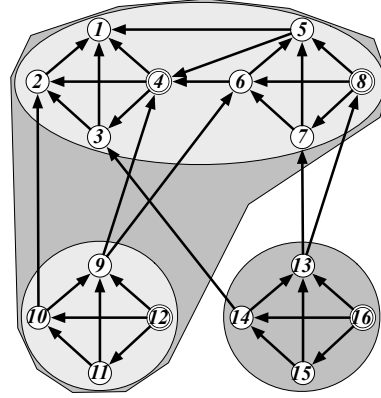


Fig. 4 The recursive construction can be applied several times to obtain additional vertices whose out-edge sets have redundancy. In this case, an out-edge from each of the 4 double-outlined vertices (4,8,12,16) can be dropped without losing persistence.

Proposition 2 gives a recursive approach for constructing persistent leader-follower formations with any desired number of vertices whose out-edge sets each contain redundancy. We refer to the leaves of the recursion as *seeds* in the construction.

Lemma 1. *Let $H = (V, E)$ be a graph resulting from any number of applications of the construction step described in Proposition 2 using formations produced by Algorithm 1 as seeds. Let R_1, \dots, R_k denote the out-edge sets of the vertices v_1, \dots, v_k with out-degree 3 and let $r_1 \in R_1, \dots, r_k \in R_k$. Then $H' = (V, E' = E - \{r_1, \dots, r_k\})$ is a persistent leader-follower formation.*

Proof. By (strong) induction on k . For the base case of $k = 1$, no applications of the construction step have occurred and H' is persistent from Algorithm 1. For the inductive step, assume the statement holds for graphs with less than K vertices of out-degree 3 and suppose H has $k = K > 1$. Then at least one construction step has occurred. Let $I = (V_I, E_I)$ and $J = (V_J, E_J)$ be the two input graphs with $V = V_I \cup V_J$ and $E \setminus (E_I \cup E_J) = \{e_1, e_2, e_3\}$, where e_1, e_2, e_3 are the edges added by the construction. Since the number of vertices of out-degree 3 in I and J must both be at least 1 and thus less than K , $I' = (V_I, E'_I = E_I \cap E')$ and $J' = (V_J, E'_J = E_J \cap E')$ are persistent leader-follower graphs by induction. Since e_1, e_2, e_3 have sources in H with out-degree exactly 2, H' is precisely $(V_I \cup V_J, E'_I \cup E'_J \cup \{e_1, e_2, e_3\})$, a persistent leader-follower formation constructed using Proposition 2. \square

Figures 3(b) and 4 depict examples of this recursive construction, using the persistent K_4 formation of Figure 2(d) as the seeds. An out-edge from each of the double-outlined vertices with out-degree 3 may be dropped without losing persistence.

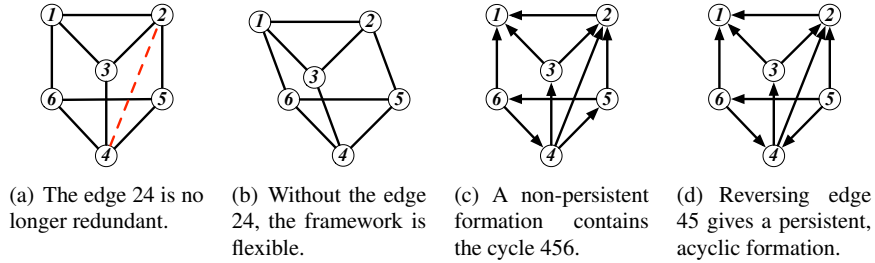


Fig. 5 A generic rigidity circuit in a special geometric position.

Acyclic persistent formations for simulation. Persistent formations without cycles are of particular interest when considering autonomous agents. We consider the situation where the leader and co-leader may move first (e.g., via tele-operation); the followers will then move to satisfy their constraints. If a directed cycle is present in the graph, the formation may not be able to converge to an embedding satisfying the constraints, even if it is persistent.

However, if a persistent graph is acyclic, then there exists an ordering of the vertices such that (1) the first vertex has out-degree 0, (2) the second out-degree 1, and (3) every other vertex has ≥ 2 out-edges to vertices earlier in the ordering [5]. Graphs of this type are a generalization of *Henneberg I graphs*, given the use of the “vertex addition” step for the followers that was first described by Henneberg for minimally rigid graphs [18, 19]. The formations in Figures 2(d), 3(b) and 4 are acyclic, certified by the orderings indicated via the vertex labels. Such an ordering permits a simple algorithm for the formation to satisfy all constraints within $O(n)$ movements, one per robot, as described in Section 5.

We conclude this section by observing that, if the seeds of the recursive construction described in Proposition 2 are acyclic, the resulting formation is also acyclic. This allows us to construct persistent acyclic formations with vertices whose out-edge sets contain redundancy. Section 5 provides simulation results that verify the ability of such formations to remain persistent even when sensing links fail.

4 Special geometric conditions

In Section 3, we presented approaches for constructing generically persistent graphs, applying to situations where agents are positioned with a generic embedding in the plane. In practice, however, agents are often placed in a systematic way so that the formation takes on a specific shape or pattern. Using a symmetric or repetitive configuration to execute certain tasks, or simply for aesthetic reasons, may lead to a non-generic embedding. In this section we discuss how such a special geometry in the formation of the agents can impact the formation’s redundancy and persistence.

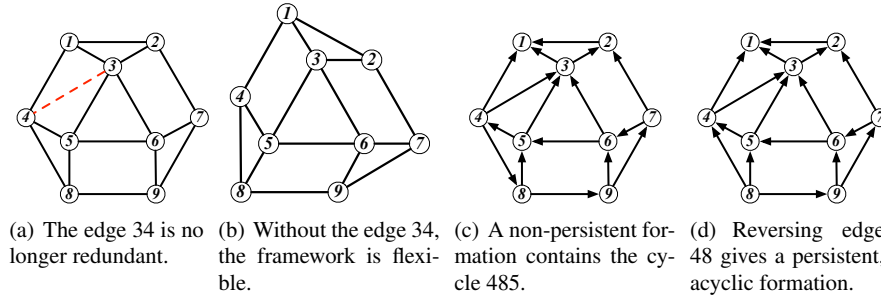


Fig. 6 A generic rigidity circuit in a special geometric position.

As noted in Section 2, for particular geometric configurations, a generically rigid graph may become flexible. Such configurations, however, are in general difficult to detect; the problem of determining whether a framework is rigid is coNP-hard [1]. A highly active research area in geometric rigidity theory is to study under what conditions non-trivial symmetries in a framework lead to flexibility in a generically rigid graph. We refer the reader to [9, 13, 14] for some key results in this area.

Given a formation of autonomous agents, it follows that special geometric conditions (as induced by symmetry in the formation, for example) can affect redundancy. For the situation we study, where the underlying undirected graph is a generic rigidity circuit, a special geometric embedding may: (1) cease to be redundant, with the removal of some edge yielding a flexible graph; or (2) cease to be rigid. We illustrate these two types of special positions with some simple examples.

Figures 5 and 6 depict two examples for type 1. Each of the graphs is a generic rigidity circuit, but are in special positions that cause the dashed edges to cease to be redundant; their removal yields flexible frameworks. For the triangular prism “Desargues” graph obtained by removing the edge 24 from the graph in Figure 5(a), it is well-known (and easy to verify) that if 1346 and 2345 are parallelograms then the framework becomes flexible, as in Figure 5(b). A geometric analysis verifies that an embedding of the graph obtained by removing the edge 34 from the graph in Figure 6(a) becomes flexible if the three congruent faces 1354, 2367 and 5698 are parallelograms, as shown in Figure 6(b).

As observed in [5], the formation shown in Figure 5(c) is not persistent. This follows from the result:

Theorem 2 (Remark 2(b) of [5]). *A formation with no vertex having a position collinear with two or more of its neighbours is persistent if and only if the framework of every subgraph obtained by removing out-edges from vertices with out-degree > 2 until all vertices have out-degree ≤ 2 is rigid.*

In Figure 5(c), only vertex 4 has out-degree 3, but removing the edge $\vec{42}$ leads to a flexible framework. However, the formation in Figure 5(d) is persistent; only vertex 5 has out-degree 3 and removal of any of its edges maintains rigidity. Similarly,



(a) A special geometric embedding with points on the x -axis and y -axis. (b) In this position, the framework is flexible.

Fig. 7 The bipartite graph $K_{3,4}$, a generic rigidity circuit, is flexible in this special position.

Figure 6(c) depicts a formation that is not persistent (the only vertex of out-degree 3 is 4, and removal of the edge $\vec{43}$ gives a flexible framework), while Figure 6(d) depicts a formation that is persistent (the only out-degree 3 vertex is 8).

Figure 7 depicts an example for type 2: the complete bipartite graph $K_{3,4}$, which is again a generic rigidity circuit. It can be shown that if the vertices of each partite set are collinear and the two lines are perpendicular, then the framework becomes flexible. We note that there exists a range of further examples of generically rigid (and redundant) graphs which become flexible due to special geometric configurations. See [9, 14], for example, for situations where continuous flexibility is induced by symmetry in the framework. It follows from Theorem 2 that any formation with an underlying framework of this type will not be persistent.

Acyclic formations with underlying special positions. Observe that the persistent formations of Fig. 5 and 6 are acyclic, while the non-persistent formations are not. This is not a coincidence; removing edge 24, respectively 34, is the only way to obtain a flexible framework. Therefore, a generically persistent graph would become a non-persistent formation with these embeddings exactly when the vertex of out-degree 3 is incident to the “essential” edge. Without the essential edge, though, one can check that any generically persistent orientation contains a cycle. The same is true if we drop *any* edge in the graph in Fig. 7. However, if we restrict ourselves to acyclic leader-follower formations on generic rigidity circuits as described in Section 3, the following result shows that special positions cannot destroy persistence.

Lemma 2. *If H is an acyclic generically persistent graph with: (1) exactly one vertex of out-degree 0, (2) exactly one vertex of out-degree 1, (3) exactly one vertex of out-degree 3 and (4) an underlying undirected generic rigidity circuit, then any formation of H (having no vertex collinear with two neighbors) is persistent.*

Proof. Remove an edge from the vertex with out-degree 3 in H . Then, by Propositions 1 and 3 in [5], we obtain an acyclic generically minimally persistent graph G . As shown in [5], such graphs can be constructed from a single edge using only vertex additions, and hence any embedding of G is rigid. Moreover, every vertex of G has an out-degree of at most 2. The result now follows from Theorem 2. \square

5 Simulation

To verify the expected behavior of the approach in Section 3, we simulated three formations using Webots [17]: one produced by Algorithm 1 and depicted in Figure 2(d), one using a single construction step of Proposition 2 as per Figure 3(b), and one using 3 construction steps as depicted in Figure 4.

Setup. Each vertex in the graph is implemented by a robot that is equipped with a generic emitter, and each directed edge by equipping the source robot with a receiver that listens to the target robot’s emitter. From the strength and direction of the signal, the source robot computes the relative position of the target; these measurements are without noise. Every follower computes and moves to a goal position based on its assigned distance constraints: with two constraints, the closest point of the (generally) 2 intersection points of 2 circles, and, with three constraints, the average of 3 points computed for each pair of constraints. To simplify control of the formation, the leader and co-leader vertices are attached to a single leader robot with emitters at each location.

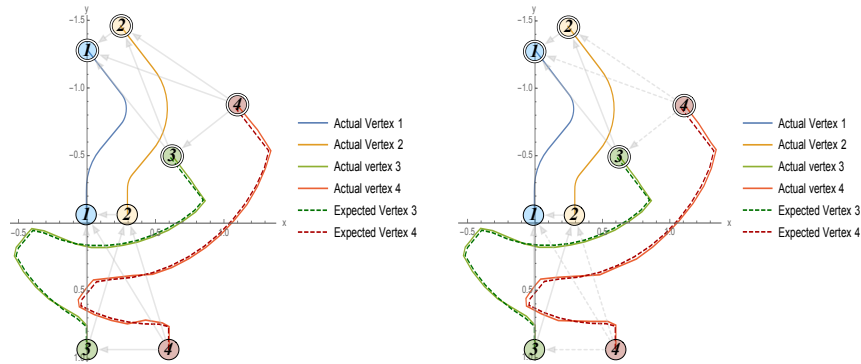
The implementation assumes an acyclic persistent formation. While the associated ordering is not explicitly used, each robot waits for a “go” signal from the 2 or 3 robots it is following. We know that each follower robot must adjust its position only once by using this approach, but we do not know a priori the time required for actuation to its goal position. Once it has moved within a specified *constraint accuracy threshold*, it emits a “go” signal that can be received by any subsequent followers. We accommodate for the unknown actuation time by moving the leader robot in steps with a delay between steps that will allow for the formation’s movements; after each such *simulation step*, we assume the formation has converged to an embedding that satisfies the constraints within the constraint accuracy threshold.

Results. As a control, one simulation was executed for each formation with all edges present. To confirm expected redundancy, an edge was randomly chosen and dropped from each robot with out-degree 3; the leader robot follows the same path as the control. Random simulations were repeated 20 times.

The expected position of the follower robots is computed using the position of the leader and co-leader positions; since the co-leader is on the same robot as the leader, its distance constraint is always satisfied. The results of the simulations are summarized in Table 1, with leader and co-leader vertices omitted from the calculations; the expected and actual paths for Formation 1 are shown in Figure 8. For each simulation step t , the difference δ_i between the expected and actual positions of follower i was computed to obtain $m_t = \min_i \delta_i$, $\mu_t = \text{mean}_i \delta_i$ and $M_t = \max_i \delta_i$ values. Data for each simulation was then computed as the mean of the m_t , μ_t and M_t across all simulation steps. The mean distance constraint length was 1.66325 ± 0.577504 meters across all formations; simulations were performed with a 0.02 meter constraint accuracy threshold. We expect the accuracy threshold to accumulate error for robots later in the sequence, as reflected in the maximum data values. The results confirm the expected behavior of the formations; the simulations testing redundancy performed comparably to the control with all edges present.

	Mean of minimums	Mean of means	Mean of maximums
Formation 1 control	0.0132058 ± 0.004	0.0173324 ± 0.006	0.021459 ± 0.010
Formation 1 random	0.0131554 ± 0.004	0.0179166 ± 0.006	0.0235468 ± 0.010
Formation 2 control	0.00826846 ± 0.004	0.0810221 ± 0.011	0.191657 ± 0.034
Formation 2 random	0.00769588 ± 0.004	0.0872681 ± 0.012	0.261483 ± 0.104
Formation 3 control	0.0087836 ± 0.004	0.0766168 ± 0.051	0.362899 ± 0.361
Formation 3 random	0.00835823 ± 0.004	0.0608315 ± 0.045	0.197745 ± 0.162

Table 1 Mean with standard deviation data for simulations (constraint accuracy threshold 0.02); for random simulations, mean values across the 20 experiments are provided.



(a) Control simulation, with all edges included. (b) Random simulation, with 1 redundant edge randomly dropped (from one of 20 simulations).

Fig. 8 Simulation paths for Formation 1, with starting (single-outlined) and ending (double-outlined) positions highlighted. Coordinates are in meters.

6 Conclusions and Future Work

This paper introduces a new class of generically persistent leader-follower graphs with redundancy for formations in the plane along with a constructive approach for generating them. The approach can be restricted to generate acyclic formations, for which a simple approach can be used to satisfy all constraints within $O(n)$ movements. Simulations verified the expected behavior of the formations when a redundant constraint was dropped.

Under special geometric conditions, certain non-generic embeddings may no longer maintain the persistence and redundancy properties; while acyclic formations are not susceptible, it remains open to characterize the behavior of persistent formations with cycles. We recognize that the redundancy property exhibited by the class of graphs presented here is quite restrictive and wish to address the strong notion of *redundantly persistent* formations defined in Section 2. Finally, our model does not incorporate noise, as it is based on equality constraints stemming from classical rigidity theory. We hope to further develop and evaluate a theoretical framework that would be robust to noise and disturbances.

Acknowledgements. We are grateful for discussions at the 2014 AIM Workshop on Configuration spaces of linkages, 2015 BIRS Workshop on Advances in Combinatorial and Geometric Rigidity and the 2016 ICMS Workshop on Geometric Rigidity Theory and Applications. We would also like to thank Joydeep Biswas for insightful conversations that stimulated the results in this paper and reviewers for their helpful feedback.

References

- [1] Abbott TG (2008) Generalizations of Kempe’s universality theorem. Master’s thesis, Massachusetts Institute of Technology, URL <http://web.mit.edu/tabbott/www/papers/mthesis.pdf>
- [2] Das AK, Fierro R, Kumar V, Ostrowski JP, Spletzer J, Taylor CJ (2002) A vision-based formation control framework. *IEEE Transactions on Robotics and Automation* 18(5):813–825
- [3] Eren T, Anderson BDO, Morse AS, Whiteley W, Belhumeur PN (2004) Operations on rigid formations of autonomous agents. *Communications in Information and Systems* 3(4):223–258
- [4] Eren T, Whiteley W, Anderson BDO, Morse AS, Belhumeur PN (2005) Information structures to secure control of rigid formations with leader-follower structure. In: *Proc. of the American Control Conference*, pp 2966–2971
- [5] Hendrickx JM, Anderson BDO, Delvenne JC, Blondel VD (2007) Directed graphs for the analysis of rigidity and persistence in autonomous agent systems. *International Journal of Robust and Nonlinear Control* 17(10-11):960–981
- [6] Hendrickx JM, Yu C, Fidan B, Anderson BDO (2008) Rigidity and persistence for ensuring shape maintenance in multiagent meta-formations. *Asian Journal of control (special issue on Collective Behavior and Control of Multi-Agent Systems)* 10(2):131–143, URL <http://arxiv.org/abs/0710.2659>
- [7] Hichri B, Adouane L, Fauroux JC, Mezouar Y, Doroftei I (2016) Cooperative mobile robot control architecture for lifting and transportation of any shape payload. *Distributed Autonomous Robotic Systems: The 12th International Symposium* pp 177–191
- [8] Jacobs D, Hendrickson B (1997) An Algorithm for Two-Dimensional Rigidity Percolation: The Pebble Game. *Journal of Computational Physics* 137(CP975809):346 – 365
- [9] Jordán T, Kaszanitzky V, Tanigawa S (2016) Gain-sparsity and symmetry-forced rigidity in the plane. *Discrete Comp Geometry* 55(3):314–372
- [10] Lee A, Streinu I (2008) Pebble game algorithms and sparse graphs. *Discrete Mathematics* 308(8):1425–1437
- [11] Olfati-Saber R, Murray RM (2002) Graph rigidity and distributed formation stabilization of multi-vehicle systems. In: *Proceedings of the IEEE Int. Conference on Decision and Control*, pp 2965–2971

- [12] Oxley JG (1992) *Matroid Theory*. Oxford University Press
- [13] Schulze B (2010) Symmetry as a sufficient condition for a finite flex. *SIAM J on Discrete Mathematics* 24(4):1291–1312
- [14] Schulze B, Whiteley W (2011) The orbit rigidity matrix of a symmetric framework. *Discrete Comp Geometry* 46(3):561–598
- [15] Vilca J, Adouane L, Mezouar Y (2016) Adaptive leader-follower formation in cluttered environment using dynamic target reconfiguration. *Distributed Autonomous Robotic Systems: The 12th International Symposium* pp 237–254
- [16] Wang Z, Schwager M (2016) Multi-robot manipulation without communication. *Distributed Autonomous Robotic Systems: The 12th International Symposium* pp 135–149
- [17] Webots (2016) <http://www.cyberbotics.com>. Commercial Mobile Robot Simulation Software
- [18] Whiteley W (1996) Some matroids from discrete applied geometry. In: Bonin J, Oxley JG, Servatius B (eds) *Matroid Theory, Contemporary Mathematics*, vol 197, American Mathematical Society, pp 171–311
- [19] Whiteley W (2004) Rigidity and scene analysis. In: Goodman J, O’Rourke J (eds) *Handbook of Discrete and Computational Geometry*, 2nd ed., Chapman & Hall/CRC, pp 1315–1254