Redundant Persistent Acyclic Formations for Vision-Based Control of Distributed Multi-Agent Formations

Alyxander Burns[*] Peter Klemperer[†] Jaemarie Solyst[‡] Audrey St. John[§]

Abstract

We present theoretical and experimental results on the application of acyclic persistent leader-follower formations with redundancy to a distributed multi-agent system. A leader-follower formation is defined on a set of point agents constrained by fixed distance assignments for following other agents; if satisfying the constraints results in the distances between all pairs of agents being maintained, the formation is *persistent*. The (generic) persistence of a leader-follower formation in 2D is combinatorially characterized by a directed graph with one "leader" vertex having no out-edges, one "co-leader" vertex having exactly one out-edge (to the leader), all other "follower" vertices having out-degree at least 2, and an underlying minimally rigid (undirected) graph.

We provide theoretical results for three types of persistent formations with redundancy, including an inductive construction for generating redundantly persistent graphs (the strongest notion of redundancy). We apply redundant persistence to multi-robot systems as a mechanism for incorporating robustness to sensing failure. In particular, we implement the approach on a vision-based distributed multi-robot platform. Using acyclic orientations permits a simple, "wave"-based control that converges reliably, and redundant edges allow the control to recover effectively from sensing limitations (e.g., a camera's limited field of view or obstruction by another robot).

1 Introduction

In applications such as collective transport [12], a formation of robots may be required to maintain a single "shape" or rigid structure. We focus on *leader-follower* formations in the plane, where two pre-specified agents determine the formation's trajectory (e.g., via teleoperation): a *leader* with two degrees of freedom and a *co-leader*[1] with one degree of freedom. The remaining robots are called *followers* and compute their trajectories independently using local geometric conditions.

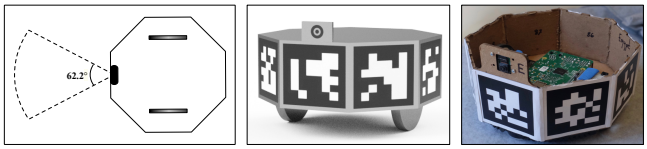


Figure 1: From left to right is a schematic of bottom view of robot (horizontal field of view is 62.2°), 3D model of robot design, picture taken of built robot without cover to show hardware inside.

By modeling these geometric constraints as directed edges, *persistence theory* [8] can be applied to distributed multi-agent systems as a way of controlling a leader-follower formation so that it maintains global structure through local sensing and actuation. Persistence theory is analogous to rigidity theory, where the combinatorics of bar-and-joint frameworks (defined by fixed distance constraints between points) can capture the rigidity or flexibility of the system. When applied to a multi-agent formation, where agents are modeled as points, the source of a directed edge can be assigned the **sensing** and **maintenance** of the distance constraint, reducing overall sensor and actuation costs.

Sensing on a mobile robotic platform is susceptible to hardware and environmental limitations (for instance, a hardware component may fail or the field of view of a camera may be obstructed or limited by lighting). To address this aspect, we build robustness into the theoretical model using constraint redundancy. While a combinatorial characterization and an efficient algorithm for redundantly rigid 2D bar-and-joint frameworks are both known, we are only aware of redundancy in persistence being studied in our previous work [3]. Furthermore, the behavior of persistent formations does not always mimic the corresponding notions in rigid frameworks. Particularly relevant to redundancy is that, while adding an edge to a rigid framework maintains rigidity, **adding an edge to a persistent formation may cause persistence to be lost** [**8**]. In [3], it was shown that every *rigidity circuit* can be oriented

[*]College of Information and Computer Sciences, UMass Amherst, alyxanderbur@cs.umass.edu, *partially supported by NSF IIS-1253146*

[†]Department of Computer Science & Engineering, Mount Holyoke College, pklemper@mtholyoke.edu

[‡]Department of Computer Science, Mount Holyoke College, solys22j@mtholyoke.edu

[§]Department of Computer Science, Mount Holyoke College, astjohn@mtholyoke.edu, *partially supported by NSF IIS-1253146*

[1]The term "first-follower" is also used in related work.

to create a persistent leader-follower formation (with a weak notion of redundancy), and a subsequent recursive algorithm was presented.

In a distributed system, each mobile robot must have a control scheme for actuation that satisfies its set of assigned distance constraints. We restrict our focus to *acyclic* persistent formations, captured by a directed graph with no cycles; a simple, "wave"-based control approach can then be used to ensure reliable convergence.

1.1 Related work.

Results from (undirected) rigidity theory have been applied in similar domains, including network localization [4], decentralized approaches to rigid network constructions [16, 19] and control of multi-robot formations in 3D [20]. A necessary condition for global rigidity is redundant rigidity (drop any edge and remain rigid), leading to the adaptation of the "pebble game" algorithm of [11] to a distributed network. In [5], redundancy is noted as an obstacle to formation control due to real-world sensing inconsistencies.

A directed application of rigidity theory to leader-follower architectures appeared in [5], with persistence theory developed in [8]. A combinatorial characterization for acyclic persistent leader-follower formations was given in [8], along with an inductive "vertex addition" construction technique; the vertex addition step is also known in the rigidity theoretic literature as a 0-extension or a Henneberg I step. The work of [15] uses this construction approach to generate minimally persistent leader-follower formations; a control scheme based on "target points" is proposed and analyzed through simulation and proof of convergence. Given an undirected graph, the existence of an acyclic persistent orientation can be checked in polynomial time [1].

We apply the concept of "cooperative positioning," where follower robots sense other neighboring robots, which are momentarily stationary and serve as "landmarks" [13] for pose calculations. Distance or range measurements can be sensed via an external motion capture system [20], a combination of GPS and computer vision [9], or even infrared [10, 18]. We use popular onboard computer vision techniques for sensing local information, allowing a completely distributed approach.

This paper builds upon the work of [3]; to the best of our knowledge, the use of redundancy as a tool for robustness had not been studied previously in the literature[2]. In this work, we address additional notions of redundancy in persistence, left open in [3], and demonstrate the application on a vision-based distributed multi-robot platform (the simulation results of [3] relied on idealized beacon sensors).

[2]Research in this area is scattered, appearing in various mathematical, computational and engineering settings.

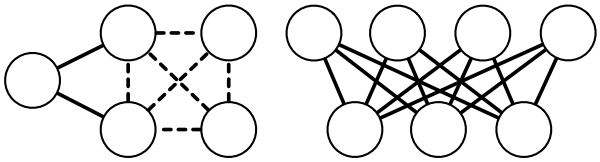


Figure 2: (a) A rigid graph that is not minimally rigid (removing a dashed edge does not result in a flexible framework) and is not redundantly rigid (removing a solid edge does result in a flexible framework). The K_4 subgraph is a rigidity circuit, as removing any edge results in a minimally rigid graph. (b) The complete bipartite graph $K_{3,4}$ is a rigidity circuit.

1.2 Contributions.

We provide theoretical and applied results for acyclic redundant persistence. Analogous to rigidity circuits, we define a *persistence circuit* to be a directed graph such that the removal of any edge results in a minimally persistent graph. We give an algorithm for constructing a persistence circuit from a rigidity circuit before proving that a persistence circuit cannot be a leader-follower formation. By relaxing the notion of redundancy to sets of redundant edges (where any edge in the set may be dropped without impacting persistence), the work of [3] gives an algorithm for constructing a persistent leader-follower formation from a rigidity circuit. In this paper, we show that not every rigidity circuit has an *acyclic* persistent leader-follower formation. Finally, we work with a strong notion of redundancy by defining *redundantly persistent* leader-follower formations, where almost any edge can be dropped without losing persistence, and give an inductive construction algorithm.

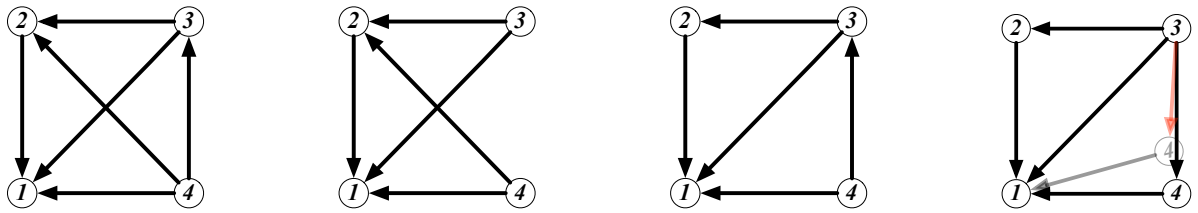
We conclude by applying the theoretical framework to a homogeneous formation of non-holonomic (differential drive) robots, each equipped with a single camera for sensing (see Figure 1). We provide experimental results that validate the ability for a fully distributed multi-robot formation to move and converge to a global structure, where redundancy provides robustness in the presence of limited sensing capabilities.

2 Preliminaries

Our work relies on persistence theory [8] and rigidity theory (see, e.g., [7, 17]). For containment, we give a high-level overview of the relevant concepts here.

2.1 Rigidity theory

Let $G = (V, E)$ be an undirected graph with n vertices and $\ell : E \to \mathbb{R}$ an assignment of distances (or lengths) to each edge; we refer to (G, ℓ) as a (bar-and-joint) *framework*. If $\mathbf{p} \in (\mathbb{R}^2)^n$ assigns positions to each vertex such that the distance constraints are satisfied,



(a) An acyclic persistent formation with redundancy; any edge from vertex 4 can be dropped without losing persistence.

(b) Without an edge, the formation from 3a is minimally persistent; its acyclicity permits an ordering with 3 waves: ({1}, {2}, {3,4}).

(c) Dropping a different edge gives a different minimally persistent formation with an ordering requiring 4 waves: ({1}, {2}, {3},{4}).

(d) The underlying (rigid) graph remains the same, but reversing edge 43 in the formation from 3c results in a formation that is not persistent.

Figure 3: Examples of persistent and not persistent graphs on 4 vertices.

i.e., $||\mathbf{p}(i) - \mathbf{p}(j)|| = l(ij)^2$ for each edge $ij \in E$, we refer to $\mathbf{p}$ as a *realization* of the framework (G, ℓ). A graph is said to be (generically[3]) *rigid* if the distances between all pairs of vertices is determined by the distances specified by edges; otherwise, it is *flexible*. A graph is *minimally rigid* if the removal of any edge results in a flexible graph. Rigid graphs that are not minimally rigid include redundant edges; such a graph is called *redundantly rigid* if the removal of *any* edge results in a rigid graph. Minimality of redundant rigidity is captured through the notion of a *rigidity circuit* (corresponding to the circuits of the rigidity matroid), a rigid graph where removing any edge results in a minimally rigid graph. Figure 2(a) depicts a rigid graph that is neither minimally rigid nor redundantly rigid; removing any dashed edge maintains rigidity, but removing a solid edge results in a flexible graph. The complete graph K_4 (dashed edges) and complete bipartite graph $K_{3,4}$ (also in Figure 2) are rigidity circuits.

Rigidity is defined via undirected graphs, so a straightforward application to a multi-robot formation could require both robots to sense and maintain the constraint dictated by an edge, incurring unnecessary sensing, computation and actuation costs. Therefore, the notion of persistence provides the analogous definition in the directed setting: a directed graph is *persistent* if each vertex can be assigned a position satisfying the distance constraints dictated by its out-going edges and the pairwise distances between all vertices is determined.

Figure 3 highlights the distinct behavior of persistence. The formations of Figures 3c and 3d share the same underlying undirected (rigid) graph, but only the formation of Figure 3c is persistent. The formation depicted in Figure 3d does not have the property that every vertex can satisfy its assigned constraints. Intuitively, vertex 4 can move anywhere on a circle about vertex 1. Vertex 3 can find a position that satisfies two of the three out-going constraints; however, for almost all positions of vertex 4, such as the faded gray position, the third constraint (red edge $\overrightarrow{34}$) will be violated.

The formal definition of persistence requires technical overhead that is outside the scope of this paper. Instead, we will use the following characterization of persistence from Theorem 3 of [8]: a directed graph is persistent if and only if the underlying undirected graph of every subgraph obtained by removing out-edges from vertices with degree ≥ 2 until all vertices have out-degree ≤ 2 is rigid. In particular, a directed graph is *minimally persistent* (i.e., removing any constraint results in a loss of persistence) if and only if its underlying undirected graph is (minimally) rigid and every vertex has out-degree at most 2. For example, the graph in Figure 3a is persistent; without an edge, as in Figures 3b and 3c, it is minimally persistent. We are interested in persistent *leader-follower* formations, where there is a "leader" vertex with out-degree 0, a "co-leader" vertex with out-degree 1 incident to the leader and all other vertices having out-degree at least 2. Since the leader and co-leader vertices have 3 degrees of freedom between them, their positions can be used to determine the coordinates of a persistent formation.

3 Results

We present results on three types of redundancy for persistence theory, analogous to those found in rigidity theory: (1) persistence circuits, (2) persistent formations with redundant edges, and (3) redundantly persistent formations. Due to space constraints, we refer the reader to the Appendix for proofs of the results (including proofs of algorithm correctness) in this section.

3.1 Persistence circuits

In rigidity theory, minimal redundancy is captured by rigidity circuits: removing any edge results in a mini-

[3]The technical definition of genericity is outside of the scope of this paper, but can be thought of as applying to "almost all" realizations (i.e., those not in a special position).

(a) First, add vertices 1 and 2 to V and edge $\overrightarrow{21}$ to A.

(b) Add vertex 3 to V and edges $\overrightarrow{31}$ and $\overrightarrow{32}$ to A.

(c) Add vertex 4 to V and edges $\overrightarrow{41}$, $\overrightarrow{42}$, and $\overrightarrow{43}$ to A.

(d) Add vertex 5 to V and edges $\overrightarrow{52}$, $\overrightarrow{53}$, and $\overrightarrow{54}$ to A.

Figure 4: Construction of an acyclic redundantly persistent formation using Algorithm 2.

mally rigid graph. Analogously, we define a *persistence circuit* to be a graph such that any edge's removal gives a minimally persistent graph.

A persistence circuit must have an underlying rigidity circuit. Since rigidity circuits are well-understood combinatorially and can be constructed inductively [2], Algorithm 1 produces a *persistence circuit* from a rigidity circuit. The $(2,3)$-pebble game algorithm of [11, 14], which appears as a subroutine, determines rigidity of an undirected graph in quadratic time by constructing a directed graph. If the input graph is rigid, the output will be a directed graph with out-degree at most 2 and an underlying undirected minimally rigid graph, i.e., a minimally persistent graph. Throughout the algorithm, pebbles are used as a mechanism for controlling the out-degree of a vertex; "pebble collection moves" rely on depth-first search to re-orient directed edges.

Algorithm 1 Construction of a persistence circuit from a rigidity circuit

Given a rigidity circuit $G = (V, E)$:

1. Remove any edge $e = ij \in E$.
2. Play the $(2,3)$-pebble game on $G' = (V, E \setminus \{e\})$ to obtain a directed graph H.
3. If i has 0 pebbles, use pebble collection moves on H to collect a single pebble on i.
4. Output the resulting directed graph with the additional edge $\overrightarrow{ij}$.

While persistent, the class of graphs output by Algorithm 1 are not leader-follower formations. In fact, we can show the following:

Lemma 1 *A persistent leader-follower formation cannot be a persistence circuit.*

Although we cannot have leader-follower formations which are persistence circuits, [3] presents an algorithm that orients a rigidity circuit to be a persistent leader-follower formation with a more restricted notion of redundancy. The graphs produced contain sets of *redundant* edges; for example, the set of out-edges from vertex 4 in Figure 3a is redundant, as any edge may be dropped without losing persistence.

3.2 Acyclic persistent formations with redundant edge sets

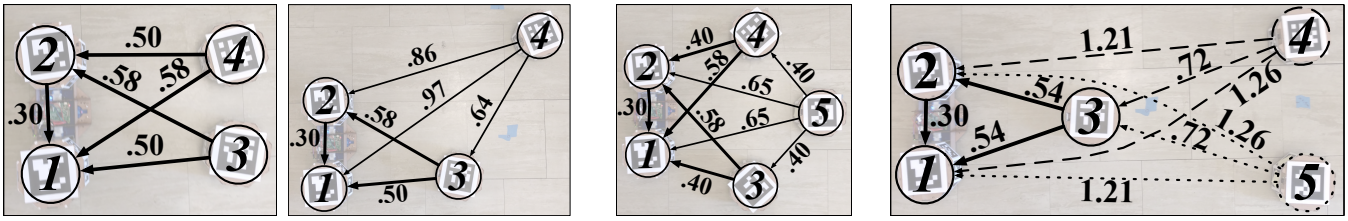
We now restrict our focus to **acyclic** persistent formations, which permit a "wave"-based control approach to satisfying the constraints. These are characterized combinatorially in Theorem 5 of [8]: an acyclic graph is persistent if and only if it has (1) one "leader" vertex with out-degree 0, (2) one "co-leader" vertex with out-degree 1, incident to the "leader," (3) all other "follower" vertices with out-degree 2 or larger. Then there exists a *Henneberg sequence* for the vertices (corresponding to an inductive "Henneberg"-type construction of the underlying undirected rigid graph) such that each vertex only has out-going edges to vertices earlier in the sequence. The first two vertices in the sequence are the leader and co-leader. We can group the subsequent vertices into a sequence of k *waves* $(w_0, w_1, \ldots, w_{k-1})$, where $w_i \subset V$, such that the vertices in a wave only have out-going edges to vertices in earlier waves. For example, Figure 3b depicts a graph with 3 waves, while Figure 3c requires 4 waves. We use the waves to control a formation so that it converges reliably, as shown through the experimental results of Section 4.

Note that the algorithm from [3] can be used to find an acyclic persistent leader-follower orientation of a rigidity circuit, if one exists, by brute-force consideration of the removal of every edge. This naturally leads to the question of whether there are rigidity circuits for which no acyclic persistent leader-follower orientations exist. The following gives the answer in the negative.

Lemma 2 *There are no acyclic leader-follower persistent orientations of the rigidity circuit $K_{3,4}$.*

3.3 Inductive constructions for acyclic redundantly persistent leader-follower formations

Following the results from the previous section, it is of interest to understand properties of acyclic persistent leader-follower formations. Theorem 5 of [8] implies that the out-edges of any follower vertex with out-



(a) Rectangle Formation (b) Kite Formation (c) Star Formation (d) Butterfly Formation

Figure 5: Screenshots of persistent formations taken from overhead during an experiment. The robot IDs and specified distance constraint (in meters) label the graph.

degree larger than 2 form a redundant set of constraints: any subset of edges such that at least 2 remain can be dropped without losing persistence. For example, consider the Butterfly graph of Figure 5d. Vertex 1 is the leader, vertex 2 the co-leader and vertices 3, 4 and 5 are followers. Since vertices 4 and 5 have out-degree 3, their out-edge sets (dashed and dotted, respectively) each form a redundant set; any out-edge can be dropped from either set without losing persistence.

Algorithm 2 constructs graphs that are acyclic persistent leader-follower formations which satisfy a version of the (strong) notion of being redundantly persistent. We define this notion in the context of an acyclic leader-follower formation, which always contains a base triangle, so we do not require redundancy within it. Let $G = (V, E)$ be an acyclic persistent leader-follower formation and $v_L, v_C, v_3, \ldots, v_{|V|}$ be a Henneberg ordering for G; let $E_b = \{\overrightarrow{v_C v_L}, \overrightarrow{v_3 v_L}, \overrightarrow{v_3 v_C}\}$ be the *base triangle* edge set. We call G *redundantly persistent* if the removal of any edge e not in the base triangle results in a persistent graph. We extend the trilateration approach of [4] to persistence to inductively construct acyclic redundantly persistent leader-follower formations. Refer to Figure 4 for an example run of the construction. Note that any set of vertices can be chosen in Step 2(b); the choice could be random or constrained by additional criteria (such as visibility determined by the geometry of a formation).

Algorithm 2 Inductive construction of a redundantly persistent acyclic formation

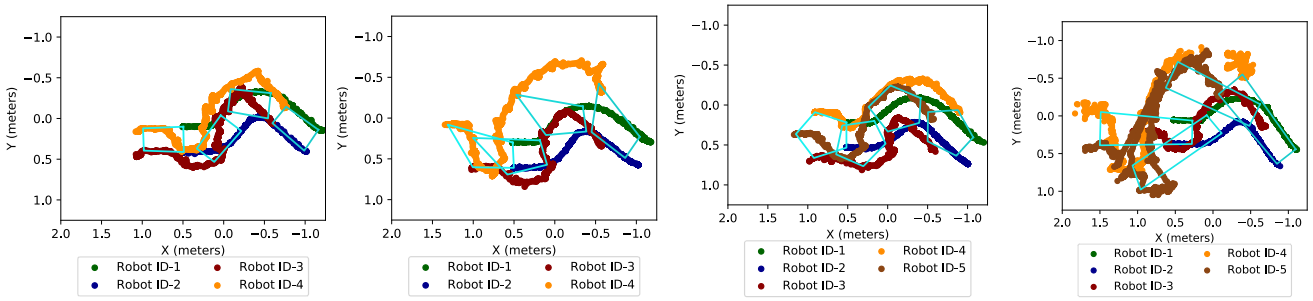
1. Initialize a set of vertices $V = \{v_L, v_C\}$ and a set of edges $A = \{\overrightarrow{v_C v_L}\}$
2. For $i \in [3..n]$:
   (a) If $i = 3$, add edges $\overrightarrow{v_i v_L}$ and $\overrightarrow{v_i v_C}$ to A
   (b) Else select a set $V' \subseteq V$ of at least 3 distinct vertices and add edges $\overrightarrow{v_i v'}$, for all $v' \in V'$, to A
   (c) Add v_i to V
3. Output the directed graph $H = (V, A)$

4 Application to multi-robot formations

Acyclic persistent leader-follower formations can be controlled with reliable convergence by having each wave (in order) sense and satisfy constraints; after the final wave is completed, the resulting positions form a realization for the underlying framework. We validate this approach through experiments on four acyclic leader-follower formations with varying levels of redundancy, depicted in Figure 5. The persistent graphs are overlaid on the multi-robot formation, with distances (in meters) labeling the edges. We chose these formations to analyze the impact of both combinatorial and geometric properties.

The Rectangle formation is minimally persistent, composed of four robots geometrically positioned at the corners of a rectangle; the two robots (vertices 3 and 4) are constrained to follow leader vertex 1 and co-leader vertex 2. The addition of edge $\overrightarrow{43}$ gives the Kite formation, where robot 4 has a set of 3 redundant out-edges; any edge may be lost without impacting the persistence of the formation. The Star formation is persistent with the out-edges of vertex 5 forming a redundant set; any pair of out-edges may be dropped without impacting persistence. Note that the Star formation is not redundantly persistent, as dropping an out-edge from vertex 4 will result in the loss of persistence. Finally, the Butterfly formation, produced by Algorithm 2 on 5 vertices, is redundantly persistent.

In all formations, vertices 1 and 2 are the leader and co-leader, together determining the global positioning of the formation. For ease of control, we simplify the implementation using a single leader robot platform containing the points for the leader and co-leader and use the same predetermined path for the leader throughout our experiments. Followers sense their positions relative to neighbors using a dictionary of OpenCV-Aruco [6] computer vision markers. When redundant edge sets are present, the follower has a prioritized list of target pairs to use for constraint maintenance. This results in a



(a) Rectangle Formation (b) Kite Formation (c) Star Formation (d) Butterfly Formation

Figure 6: Scatter plots illustrating the tracks of each robot for each of four formations. We outline the convex hull (cyan) of each formation during every tenth measurement to emphasize the formation.

more robust multi-agent system than using a minimally persistent formation, as the formation can recover from sensing failure, e.g., due to environmental factors, such as lighting, or occlusion of targets by other robots or objects. Given the relative positions of two targets, the follower computes a point satisfying both constraints (choosing the closer point of intersection of the two corresponding circles) and moves directly to that position. We refer the reader to the Appendix for more details on the hardware setup and wave-based control approach.

4.1 Experimental results

An overhead camera was used to obtain the coordinates of each robot for analysis. Figure 6 illustrates the movement tracks of the robots in each of the tested formations. Note that, for the Butterfly formation, the initial sharp turn of the leader caused Robot ID-5 to move out of the view of the overhead camera. We overlay the convex hull at timed intervals to highlight the maintenance of the overall structure.

Formation	From	To Robot 1	To Robot 2	To Robot 3
Rectangle	Robot 3	0.012±0.020, 0.168	0.004±0.019, 0.136	NA
	Robot 4	**0.009±0.012 ≤ 0.163**	**0.005±0.012, 0.166**	(-0.027±0.009 ≤ 0.057)
Kite A	Robot 3	0.005±0.008 ≤ 0.142	-0.001±0.008 ≤ 0.097	NA
	Robot 4	**0.005±0.012 ≤ 0.130**	**-0.001±0.013, 0.122**	-0.020±0.010 ≤ 0.145
Kite B	Robot 3	0.008±0.015, 0.161	0.004±0.019 ≤ 0.230	NA
	Robot 4	**0.007±0.014 ≤ 0.149**	0.001±0.013 ≤ 0.194	**-0.018±0.018 ≤ 0.139**
Kite C	Robot 3	0.005±0.019≤ 0.136	-0.002±0.018 ≤ 0.142	NA
	Robot 4	0.007±0.028 ≤ 0.181	**-0.014±0.032 ≤ 0.206**	**0.004±0.025 ≤ 0.132**

Table 1: Constraint accuracy for four formations: the Rectangle formation and three variations of Kite.

For the Rectangle and Kite formations, Table 1 provides more precise validation of constraint maintenance. Since the Kite formation has a set of redundant edges from vertex 4, we ran 3 experiments, varying the priority list of pairs of vertices to follow: Kite A used ({1,2},{1,3},{2,3}), Kite B used ({1,3},{2,3},{1,2}) and Kite C used ({2,3},{1,3},{1,2}). The results were compiled by calculating the mean error for each pair-wise constraint within each formation, then calculating the minimum, mean and maximum of those means; they are formatted as: *mean error during camera wave (m)* $\pm$*standard deviation* $\leq$ *max error from constraint.* The maximum single-wave leader movement is 0.112 m and 10 degrees, which affects the maximum error from constraint throughout the movement waves. The results in parentheses were not directly controlled, but instead maintained by the formation. Additional results and data may be found in the Appendix.

5 Remarks

Rigidity and persistence theory are typically applied as static analysis tools, but control of a multi-agent mobile formation must include approaches for dynamic movement. By working with acyclic formations, we are able to exploit Henneberg sequences (developed primarily as a tool for inductive proof techniques) to implement a wave-based control scheme with reliable convergence.

Performing experiments through a low-cost hardware platform can give insight into the challenges to creating a robust control for multi-robot formations; popular vision-based sensing is susceptible to geometric and environmental factors, including viewing angle, lighting and occlusion. Our results confirmed the robustness brought to the system through the incorporation of redundancy in the theoretical model, allowing the control to recover from sensing loss.

While redundancy in rigidity theory is well-understood, the analogous concepts in persistence theory have not been well-studied. Our results reveal distinct behaviors of redundancy in the directed graphs of persistence theory that do not arise in the undirected graphs of rigidity theory.

We are grateful to the anonymous reviewers for their detailed and helpful comments.

References

[1] J. Bang-Jensen and T. Jordan. On persistent directed graphs. *Networks*, 52:271–276, 2008.

[2] A. R. Berg and T. Jordán. A proof of connelly's conjecture on 3-connected circuits of the rigidity matroid. *J. Comb. Theory Ser. B*, 88(1):77–97, May 2003.

[3] A. Burns, B. Schulze, and A. St. John. Persistent multi-robot formations with redundancy. In R. Groß, A. Kolling, S. Berman, E. Frazzoli, A. Martinoli, F. Matsuno, and M. Gauci, editors, *Distributed Autonomous Robotic Systems: The 13th International Symposium*, pages 133–146. Springer International Publishing, 2018.

[4] T. Eren, O. K. Goldenberg, W. Whiteley, Y. R. Yang, A. S. Morse, B. D. O. Anderson, and P. N. Belhumeur. Rigidity, computation, and randomization in network localization. In *IEEE INFOCOM 2004*, volume 4, pages 2673–2684 vol.4, 2004.

[5] T. Eren, W. Whiteley, and P. N. Belhumeur. Rigid formations with leader-follower architecture. Technical report, Columbia University, CUCS-010-05, March 2005.

[6] S. Garrido-Jurado, R. M. noz Salinas, F. Madrid-Cuevas, and M. Marín-Jiménez. Automatic generation and detection of highly reliable fiducial markers under occlusion. *Pattern Recognition*, 47(6):2280 – 2292, 2014.

[7] J. Graver, B. Servatius, and H. Servatius. *Combinatorial Rigidity*, volume 2 of *Graduate Studies in Mathematics*. American Mathematical Society, 1993.

[8] J. M. Hendrickx, B. D. O. Anderson, J.-C. Delvenne, and V. D. Blondel. 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, 2007.

[9] K. Hou, H. Sun, Q. Jia, and Y. Zhang. An autonomous positioning and navigation system for spherical mobile robot. *Procedia Engineering*, 29:2556 – 2561, 2012. 2012 International Workshop on Information and Electronics Engineering.

[10] R. Islam, I. Mobin, M. N. Shakib, and M. M. Rahman. An approach of ir-based short-range correspondence systems for swarm robot balanced requisitions and communications. *CoRR*, abs/1608.03610, 2016.

[11] D. Jacobs and B. Hendrickson. An Algorithm for Two-Dimensional Rigidity Percolation: The Pebble Game. *Journal of Computational Physics*, 137(CP975809):346 – 365, 1997.

[12] C. R. Kube and E. Bonabeau. Cooperative Transport By Ants and Robots. Working Papers 99-01-008, Santa Fe Institute, Jan. 1999.

[13] R. Kurazume, S. Nagata, and S. Hirose. Cooperative positioning with multiple robots. In *Proceedings of the 1994 IEEE International Conference on Robotics and Automation*, pages 1250–1257 vol.2, 5 1994.

[14] A. Lee and I. Streinu. Pebble game algorithms and sparse graphs. *Discrete Mathematics*, 308(8):1425–1437, 2008.

[15] S. Mou, M. Cao, and A. Stephen Morse. Target-point formation control. *Automatica*, 61(C):113–118, Nov. 2015.

[16] A. Priolo, R. K. Williams, A. Gasparri, and G. S. Sukhatme. Decentralized algorithms for optimally rigid network constructions. In *IEEE International Conference on Robotics and Automation*, pages 5010–5015, 2014.

[17] M. Sitharam, A. St. John, and J. Sidman, editors. *Handbook of Geometric Constraint Systems Principles*. Chapman and Hall/CRC, 2018.

[18] H. Wei, N. Li, M. Liu, and J. Tan. A novel autonomous self-assembly distributed swarm flying robot. *Chinese Journal of Aeronautics*, 26(3):791 – 800, 2013.

[19] R. K. Williams, A. Gasparri, M. Soffietti, and G. S. Sukhatme. Redundantly rigid topologies in decentralized multi-agent networks. In *IEEE Conference on Decision and Control*, pages 6101–6108, 2015.

[20] D. Zelazo, A. Franchi, H. H. Bülthoff, and P. R. Giordano. Decentralized rigidity maintenance control with range measurements for multi-robot systems. *International Journal of Robotics Research*, 34:128, 01 2015.

Appendix

5.1 Proofs for theoretical results

We include here the proofs for correctness of Algorithm 1, Lemma 1, Lemma 2 and Algorithm 2.

Proof. [of correctness of Algorithm 1] Let $H = (V, A)$ be the output of Algorithm 1. By construction and Invariant (1) of Lemma 10 of [14] (for any vertex v, the sum of the number of pebbles on v and the out-degree of v is 2), H has either 1 vertex with out-degree 0 or 2 vertices with out-degree 1; all other vertices have out-degree 2. We show that H is a persistence circuit. Let $e \in A$ and $J = (V, A \backslash e)$. Then J either has (1) 3 vertices with out-degree 1, and all other vertices with out-degree 2; or (2) 1 vertex with out-degree 0, 1 vertex with out-degree 1, and all others with out-degree 2. Since the underlying (undirected) graph of H is a rigidity circuit, the underlying graph of J is minimally rigid. Thus, by Theorem 3 of [8], J is minimally persistent and H is a persistence circuit. □

Proof. [of Lemma 1] Let $H = (V, A)$ be a persistent leader-follower formation with v_C, v_L the leader and co-leader vertices having out-degree 0 and 1, respectively. Then there must be exactly one vertex x with out-degree 3 and all other vertices with out-degree 2. Assume, for a contradiction, that H is a persistence circuit; since the underlying graph of H must be a rigidity circuit, $|V| \geq 4$. Let $y \in V$ be a vertex with out-degree 2 and $e = \overrightarrow{yz}$ be one of the out-going edges. By assumption, $J = (V, A \setminus e)$ is minimally persistent and its underlying graph is minimally rigid. Since x has out-degree 3, dropping any out-going edge must result in a persistent graph J' by Theorem 3 of [8]. However, the underlying graph of J becomes flexible after the removal of any edge, so J' cannot be persistent, giving the contradiction. □

Proof. [of Lemma 2] Consider any persistent leader-follower orientation $G = (U, V, E)$ of $K_{3,4}$, where $|U| = 3$ and $|V| = 4$ are disjoint vertex sets. We show it must have a cycle. Suppose, for a contradiction, it is acyclic. Since $K_{3,4}$ is a rigidity circuit, $|E| = 2n - 2$ edges, there must be one vertex v_L with out-degree 0, one vertex v_C with out-degree 1, one vertex x with out-degree 3 and all other vertices with out-degree 2. Since G is an acyclic persistent orientation, there exists an ordering of vertices $v_L, v_C, v_3, \ldots, v_7$ such that the out-edges of each v_i are directed to vertices with smaller indices. Then $x = v_7 \in V$ and $v_3, \ldots, v_6$ must have out-degree 2. However, v_6 is incident to 3 vertices of the set $\{v_L, v_c, v_3, \ldots, v_5\}$ and must be oriented towards them; thus, v_6 has out-degree 3, giving the contradiction. □

Proof. [of Algorithm 2] Let G be the output of an execution of Algorithm 2. By construction, G is an acyclic graph with (1) one "leader" vertex with out-degree 0, (2) one "co-leader" vertex with out-degree 1, incident to the "leader," (3) all other "follower" vertices with out-degree 3. By Theorem 5 of [8], G is persistent. Let $e \notin E_b = \{\overrightarrow{v_C v_L}, \overrightarrow{v_3 v_L}, \overrightarrow{v_3 v_C}\}$; then e is an out-edge from a vertex of degree 3. By Theorem 3 of [8], the graph obtained by removing e is persistent. Thus, G is redundantly persistent. □

Formation	{1,2}	{1,3}	{2,3}	Search
Kite A	**98%**	0%	0%	2%
Kite B	0%	**99%**	1%	0%
Kite C	0%	76%	**23%**	1%
Star Robot 3	95%	NA	NA	5%
Star Robot 4	100%	NA	NA	0%
Star Robot 5	**96%**	0%	0%	4%
Butterfly Robot 3	100%	NA	NA	0%
Butterfly Robot 4	**4%**	71%	$< 1\%$	25%
Butterfly Robot 5	**7%**	1%	75%	17%

Table 2: Summary of the landmark-pair selections taken by robots during experiments. Followers require two landmarks and choose from a prioritized list of pairs (column labels) when more are available; the top priority pair is boldface in each row.

5.2 Experimental details

This section contains a more detailed description of our experimental setup as well as additional data and analysis.

Setup. Figure 5 provides a depiction of the hardware design. Robots are based on a Pololu Romi[4] wheeled differential-drive chassis and electronics fitted within a octagon camera target body. The leader is fitted with two camera targets 30 cm apart, centered over the Romi Chassis. On-board computation is provided by a Raspberry Pi version 3 with connected Raspberry Pi Camera (v2, 8 MP version). These robots are relatively low-cost, with a total bill-of-materials under $200 USD.

The robot shells are designed as an octagonal prism, with 2.7" edges, and 2.3" square Aruco vision markers centered within each face; a unique marker was assigned to each robot, allowing it to be used as an identifier. We chose 8 sides as it experimentally produced the best results. Shells with fewer sides were more susceptible to marker identification loss due to viewing angle or occlusion factors. Using more sides (while keeping the overall shell dimensions constant) forced a decrease in the size of the markers, resulting in marker recognition loss susceptible to distance factors; the octagonal prism allowed recognition up to 3 meters away.

Formation control. The overall movement of the formation is dictated by the leader robot; individual follower robots move to maintain their specified constraints. By working with acyclic persistent formations, we can use a simple "wave" control that converges reliably. Robots only move during their assigned waves, with the specific wave assigned to a robot using a Henneberg ordering for the formation. Each wave ends only when all robots assigned to that wave have finished moving and satisfied their constraints.

Separating the formation into waves creates several advantages beyond reliable convergence: the landmarks (Aruco markers) tracked by the followers are not moving, resulting in improved camera accuracy, followers need only minimize

[4]Pololu Corporation: https://www.pololu.com/category/203/romi-chassis-kits

constraint deviation with their landmarks, and fewer robots moving reduces the risk of collisions. However, the wave-motion procedure suffers from the disadvantage that the minimum deviation from the range-constraints is bounded by the distance traveled by the leader, and the cascading effects as the following waves move increase the error.

Redundancy data and analysis. Table 2 summarizes the landmark usage of each robot in determining movement targets throughout 5 experiments on the formations with redundancy: 3 with the Kite formation (varying the top landmark-pair in the priority list), one for the Star and one for the Butterfly formation. Each robot made approximately one-hundred movements, and the results are reported as a percentage of total movements (including those required rotate while searching for markers). Note that sensing constraints play a role in the experiment for Kite C, where (Robot 2,Robot 3) is the prioritized pair, but (Robot 1,Robot 3) determines 76% of the movements. The geometry of the formation places landmarks Robot 2 and Robot 3 at the extreme opposite edges of Robot 4's field-of-vision, making them less likely to be viewed simultaneously.