Persistence Acquisition and Maintenance for Autonomous Formations

# Changbin Yu, Barış Fidan, Brian D.O. Anderson

National ICT Australia Ltd. and
Research School of Information Sciences & Engineering,
the Australian National University,
216 Northbourne Ave., Canberra ACT 2601 Australia
{brad.yu,baris.fidan,brian.anderson}@nicta.com.au

Abstract

Built upon a recently developed theoretical framework, we consider some practical issues raised in autonomous multi-agent formation control. We use the notions of rigidity and persistence to analyze the cohesiveness of the formation structure and maintenance of the constraints on each individual agent. We present basic properties of persistent formations and give an operational criterion to determine if a formation is persistent. Using the framework we develop, we seek to provide systematic ways of acquiring persistence for classes of formations often found in real world applications. Decentralized automatic acquisition of persistence for rigid formations is an open problem and answers found for these specific examples may serve as hints for seeking a general solution. Another focus of this paper is on how to transfer autonomy (abstracted as degrees of freedom) among agents, when the formation changes with new agent(s) added, to preserve persistence.

1. INTRODUCTION

Multi-agent systems have attracted considerable attention recently as witnessed by explosion of papers in the area (see for example [1]–[5]). Agents, abstracted as vertices of graphs in this paper (following [6], [7]), can be thought as any autonomous agents including sensor agents, robots, underwater vehicles, unmanned aerial vehicles, and ground vehicles.

A formation is a collection of agents moving in real 2- or 3-dimensional space to fulfill certain mission requirements. A formation is called rigid if the distance between each pair of agents does not change over time, i.e., agents move as a cohesive whole. In a typical formation, sensing/communication links are used for maintaining fixed distances between agents. The interconnection structure of these sensing/communication links can be abstracted using a graph, namely, the underlying graph of the formation.

There are two types of control structures that can be used to maintain the required distance between pairs of agents in a formation: symmetric control and asymmetric control. In the symmetric case, to keep the distance between, e.g., agent 1 and agent 2, there is a joint effort of both agent 1 and agent 2 to simultaneously and actively maintain their relative positions. The associated graph will have an undirected edge between vertices 1 and 2. If enough agent pairs explicitly maintain distances, all inter-agent distances will be maintained and the formation will be rigid. In contrast, in the asymmetric case, only one of the agents, e.g., agent 1 actively maintains its position relative to agent 2 using the directional link from 1 to 2. This means that only agent 1 has to receive the position information broadcasted by agent 2, or sense the position of agent 2 and make decisions on its own. Therefore, in the later case, both the overall control complexity and the communication complexity (in terms of message sent or information sensed) for the formation are reduced by half.

In the recent control literature, the characterization of a point-agent formation with asymmetric control structure, which can naturally be represented by a directed graph, has started to be attempted using the notion of rigidity of a directed graph [1], [6], which is called persistence of a directed graph [7] as well. In this paper, we prefer to use the term persistence in order to distinguish it from the undirected graph notion of rigidity.

In Section 2, we formally introduce the definition of persistence (as opposed to rigidity) of formations or their underlying graphs. We present basic properties of persistent formations and give an operational criterion to determine if a formation is persistent. We show that a graph is persistent if and only if it is rigid and constraint consistent. In Section 3, we seek to provide a systematical way of acquiring persistence for classes of planar formations often found in real world applications. Decentralized automatic acquisition of persistence for rigid formations is an open problem and these examples may serve as hints for seeking a solution. In Section 4, we focus on how to transfer autonomy (abstracted as degrees of freedom) between agents, when the formation changes with new agent(s) added, to preserve persistence. We end the paper with concluding remarks in Section 5.

2. RIGIDITY AND PERSISTENCE

In this section, we review the notions of rigidity and persistence given in [7], [8], seeking to provide a theoretical
framework for real world applications. Persistence has the following intuitive meaning: A formation is persistent if, provided that all the agents tasked with satisfying certain distance constraints, the global structure of the formation is preserved, i.e., when the formation moves, it necessarily moves as a cohesive whole.

We will see that rigidity of the underlying undirected graph of a formation is a necessary but not sufficient condition for persistence. This will lead us to the notion of constraint consistence of graph, which is the additional condition for a rigid graph to be persistent. Intuitively, a graph is constraint consistent if every agent is able to satisfy all its distance constraints provided that all the others are trying to do so.

In the last subsection, we examine the properties of persistent formations. We present some fundamental results related to persistence characterization, which are used in later sections to check/verify persistence.

Note that although the following definitions, theorems and other results can be formulated for arbitrary dimensional space $\mathbb{R}^d (d \in \{2, 3, \ldots \})$, we only consider $d = 2$ or 3 in this paper for real world applications. This applies to Sections 3 and 4 as well.

A. Rigidity

In $\mathbb{R}^d (d \in \{2, 3\})$, a representation of an undirected graph $G = (V, E)$ with vector set $V$ and edge set $E$ is a function $p : V \rightarrow \mathbb{R}^d$. We say that $p(i) \in \mathbb{R}^d$ is the position of the vertex $i$, and define the distance between two representations $p_1$ and $p_2$ of the same graph by

$$d(p_1, p_2) = \max_{i \in V} ||p_1(i) - p_2(i)||.$$  

A distance set $d$ for $G$ is a set of distances $d_{ij} > 0$, defined for all edges $(i, j) \in E$. A distance set is realizable if there exists a representation $p$ of the graph for which $||p(i) - p(j)|| = d_{ij}$ for all $(i, j) \in E$. Such a representation is then called a realization. Note that each representation $p$ of a graph induces a realizable distance set (defined by $d_{ij} = ||p(i) - p(j)||$) for all $(i, j) \in E$, of which it is a realization.

A representation $p$ is rigid if there exists $\epsilon > 0$ such that for all realizations $p'$ of the distance set induced by $p$ and satisfying $d(p, p') < \epsilon$, there holds $||p(i) - p'(j)|| = ||p(i) - p(j)||$ for all $i, j \in V$ (We say in this case that $p$ and $p'$ are congruent). A graph is said to be generically rigid if almost all its representations are rigid. Some discussions on the need for using "generic" and "almost all" can be found in [7], [9]. One reason for using these terms, in $\mathbb{R}^d (d \in \{2, 3\})$, is to avoid the problems arising from having $d + 1$ or more vertices lying in a $d_1$-dimensional space where $d_1 \leq d - 1$.

A widely used notion in a rigidity analysis is minimal rigidity. A graph is called minimally rigid if it is rigid and if there exists no rigid graph with the same number of vertices and a smaller number of edges. Equivalently, a graph is minimally rigid if it is rigid and if no single edge can be removed without losing rigidity. These two equivalent definitions of minimal rigidity lead to the following theorem, which is a corollary of the results in [8].

Theorem 1: If a graph $G = (V, E)$ in $\mathbb{R}^d (d \in \{2, 3\})$ with at least $d$ vertices is rigid, then there exists a subset $E'$ of edges such that $G' = (V, E')$ is minimally rigid. This also implies the following:

- $|E'| = |V| - d + 1/2$.
- Any subgraph $G'' = (V'', E'')$ of $G'$ with at least $d$ vertices satisfies $|E''| \leq |V''| - d + 1/2$.
- Any subgraph $G'' = (V'', E'')$ of $G'$ such that $|E''| = |V''| - (d + 1)/2$ is minimally rigid.

B. Persistence

As mentioned above, rigidity is a notion examinable using undirected graphs, and to properly characterize persistence, certain notions of directed graphs are also needed. In order to have a formal definition of persistence, we first need to characterize the fact that each agent is trying to keep the distances from its neighbors constant.

Let us thus fix a directed graph $G$, desired distances $d_{ij} > 0$ for $\forall (i, j) \in E$, and a representation $p$. We say that the edge $(i, j) \in E$ is active if $||p(i) - p(j)|| = d_{ij}$. We also say that the position of the vertex $i \in V$ is fitting for the distance set $d$ if it is not possible to increase the set of active edges leaving $i$ by modifying the position of $i$ while keeping the positions of the other vertices unchanged. More formally, given a representation $p$, the position of vertex $i$ is fitting if there is no $p^* \in \mathbb{R}^d$ for which the following strict inclusion holds:

$$\{(i, j) \in E : ||p(i) - p(j)|| = d_{ij}\} \subset \{(i, j) \in E : ||p^*(i) - p^*(j)|| = d_{ij}\}$$  

This condition intuitively means that the agent $i$ cannot satisfy additional distance constraints without breaking some that it already satisfies, as shown in the two-dimensional example in Figure 2, which is drawn from [7]. A representation of a graph is a fitting representation for a certain distance set $d$ if all the vertices are at fitting positions for $d$. Note that any realization is a fitting representation for its distance set.

We can now give a formal definition of persistence: A representation $p$ is persistent if there exists $\epsilon > 0$ such that every representation $p'$ fitting for the distance set induced by
p and satisfying $d(p, p') < \epsilon$ is congruent to $p$. A graph is then generically persistent if almost all its representations are persistent.

Next, we show that a generically persistent graph in $\mathbb{R}^d (d \in \{2, 3\})$ is always generically rigid, and give a necessary and sufficient condition for a generically rigid graph to be generically persistent. This condition is called the generic constraint consistence of a graph. A representation $p$ is constraint consistent if there exists $\epsilon > 0$ such that any representation $p'$ fitting for the distance set $d$ induced by $p$ and satisfying $d(p, p') < \epsilon$ is a realization of $d$. Intuitively, the constraint consistence of a representation means that if each agent tries to satisfy its distance constraints (i.e., is at a fitting position), then all the distance constraints will be satisfied, or equivalently, no agent will be in a situation where it cannot satisfy some constraint. The illustration of such a situation in $\mathbb{R}^2$ can be found in [7]. Again, we say that a graph is generically constraint consistent if almost all its representations are constraint consistent.

We have the following useful equivalences for directed graphs in any $d$-dimensional space $\mathbb{R}^d (d \in \{2, 3\})$:

**Theorem 2:** [7] A representation is persistent if and only if it is rigid and constraint consistent. A graph is generically persistent if and only if it is generically rigid and generically constraint consistent.

**C. Characterization Of Persistent Formations**

We begin characterization of formations with underlying persistent graphs by giving a lower bound on the number of active edges, and a sufficient condition for a graph to be constraint consistent. In the sequel, $d_{\text{in}}(i)$ and $d_{\text{out}}(i)$ designate respectively the in- and out-degree of the vertex $i$ in the graph $G$. When no confusion is possible about the graph, we will use $d^-(i)$ and $d^+(i)$. Note also that the results given in this section are a summary of those presented in [7], [8].

**Lemma 1:** Let $p$ be a representation of a graph $G = (V, E)$ in $\mathbb{R}^d (d \in \{2, 3\})$, and $i$ a vertex of this graph. If the position $p(i)$ does not lie on any $(d - 1)$-dimensional hyper-plane containing $d$ or more of its neighbors, then there exists $\epsilon > 0$ such that in every representation $p' \in B(p, \epsilon)$ (i.e., such that $d(p, p') < \epsilon$) fitting for the distance set induced by $p$, the number of active edges leaving $i$ is at least $\min (d, d^+(i))$. Consequently, a graph in which all the vertices have an out-degree smaller than or equal to $d$ is always constraint consistent.

**Proposition 1:** A persistent graph in $\mathbb{R}^d (d \in \{2, 3\})$ remains persistent after deletion of any edge $(i, j)$ for which $d^+(i) \geq d + 1$. Similarly, a constraint consistent graph in $\mathbb{R}^d (d \in \{2, 3\})$ remains constraint consistent after deletion of any edge $(i, j)$ for which $d^+(i) \geq d + 1$.

An interesting corollary of Proposition 1 concerns the total number of degrees of freedom (DOFs), which we also call the total DOF count, in the graph. The number of degrees of freedom (DOF count) of a vertex is the maximal dimension, over all representations of the graph, of the set of possible fitting positions for this vertex. For example, in $\mathbb{R}^2$, the vertices with zero out-degrees have two DOFs, the vertices with out-degrees 1 have one DOF, and the others have zero DOF. Note that a vertex with zero DOF can have more than one possible fitting position. Observe indeed that, in almost all situations in $\mathbb{R}^d$, there are two possible fitting positions for a vertex with out-degree $d$. However, since this set contains a finite number of points, its dimension is still 0. The following result provides a natural bound on the total DOF count of a persistent graph.

**Corollary 1:** The total DOF count of a persistent graph in $\mathbb{R}^d (d \in \{2, 3\})$ can at most be $d(d + 1)/2$.

We have stated in Proposition 1 that a persistent graph remains persistent after deletion of any edge $(i, j)$ for which $d^+(i) \geq d + 1$. After successive deletions, we can reach in this way a persistent graph whose vertices all have an outgoing degree that is smaller than or equal to $d$. The next theorem, which is analogous to Theorem 3 in [7] states that a graph is persistent if and only if all the graphs obtained in this way are rigid in $\mathbb{R}^2$.

**Theorem 3:** A $d$-dimensional $(d \in \{2, 3\})$ graph is persistent if and only if all those subgraphs are rigid which are obtained by successively removing outgoing edges from vertices with out-degree larger than $d$ until all such vertices have an out-degree equal to $d$.

Theorem 3 provides a non-polynomial time algorithm to check the persistence of any $d$-dimensional graph for $d \in \{2, 3\}$: It is sufficient to check the rigidity of all subgraphs obtained by deleting the edges leaving vertices with out-degree larger than or equal to $d + 1$ until all the vertices have an out-degree less or equal to $d$. An algorithm with a smaller complexity would be useful in the case of large graphs, especially if there is a high number of vertices with high out-degrees, but no such algorithm is currently available. Given this difficulty of characterization of persistence, in the next section, we turn our attention to the problem of systematic construction of provably persistent planar formations by assigning directions to the links.
3. **Persistence Acquisition for Formations**

In this section, we study how formations with special underlying rigid graphs can acquire the persistence systematically. By doing so, it will reduce the control complexity by at least half as compared to the undirected version, since only one of each pair of agents is required to explicitly maintain the inter-agent distance. Again, for formal analysis, we abstract the rigid planar formations as the underlying rigid graphs, and each directed edge \((i, j)\) corresponds to agent \(i\) actively maintaining its distance from agent \(j\).

Complete graphs model the concept of closed neighborhood of UAVs flying together, where the sensing (communication) radius of each agent potentially allows it to maintain its distance actively from any other agent in the entire neighborhood. One might use wheel graphs to model the formation of which there is a central UAV, the commander, whose sensing region covers all other agents. \(C^2\) and \(C^3\) graphs illustrate the scenarios when the sensing (communication) region of each agent is doubled and tripled, respectively, so that extra scenarios when the sensing (communication) radius of \(C\) distance actively from any other agent in the entire neighborhood potentially allows it to maintain its persistence in each of these cases. The following propositions establish some systematic ways of acquiring persistence for these special classes of formations; their proofs will appear in an extended version of the paper, which is under preparation.

**Proposition 2:** Given an integer \(k \geq 3\), consider the \(k\)-complete (undirected) graph \(K_k\) in \(\mathbb{R}^d\) (\(d \in \{2, 3\}\)). Let \(\overrightarrow{K}_k\) be the directed graph obtained by assigning directions to the edges of \(K_k\) such that for any \(1 \leq i < j \leq k\), direction of edge \((i, j)\) is from \(j\) to \(i\). Then, \(\overrightarrow{K}_k\) is persistent.

**Proposition 3:** Given an integer \(k \geq 3\), consider the wheel graph \(W_k\) in \(\mathbb{R}^d\) that is composed of \(k\) rim vertices, labelled vertices \(1, 2, \ldots, k\), the rim cycle of edges \(C_k = \{(1, 2), (2, 3), \ldots, (k−1, k), (k, 1)\}\) passing through these vertices, one hub vertex (labelled vertex 0) and the edges \((0, i)\) for \(i = 1, 2, \ldots, k\) connecting the hub vertex to each of the rim vertices. Let \(\overrightarrow{W}_k\) be the directed graph obtained by assigning directions to the edges of \(W_k\) such that the direction of each rim edge \((i, i+1)\) is from \(i\) to \(i+1\), the direction of \((1, k)\) is from \(k\) to \(1\), and the direction of any edge \((0, i)\) is from \(i\) to \(0\). Then, \(\overrightarrow{W}_k\) is persistent.

**Proposition 4:** Given an integer \(k \geq 3\), consider the graph \(C^2(k)\) in \(\mathbb{R}^d\) that is composed of vertices \(0, 1, \ldots, k−1\) and edges \((i, (i+j) \mod k)\) for \(i = 0, 2, \ldots, k−1\) and \(j = 1, 2\). Let \(\overrightarrow{C}^2(k)\) be the directed graph obtained by assigning directions to the edges of \(C^2(k)\) such that direction of each edge \((i, (i+j) \mod k)\) is from \(i\) to \((i+j) \mod k\). Then, \(\overrightarrow{C}^2(k)\) is persistent.

**Proposition 5:** Given an integer \(k \geq 3\), consider the graph \(C^3(k)\) in \(\mathbb{R}^d\) (\(d \in \{2, 3\}\)) that is composed of vertices \(0, 1, \ldots, k−1\) and edges \((i, (i+j) \mod k)\) for \(i = 0, 2, \ldots, k−1\) and \(j = 1, 2, 3\). Let \(\overrightarrow{C}^3(k)\) be the directed graph obtained by assigning directions to the edges of \(C^3(k)\) using the following procedure:

1. For the subgraph of \(C^3(k)\) that is composed of vertices \(1, 2, 3, 4, 6\) and the edges among them, which is a complete graph, assign directions according to Proposition 2.
2. For vertices \(i = 5, 6, \ldots, k−3\) assign the directions of the edges \((i, i−1), (i, i−2)\) and \((i, i−3)\) such that all three directed edges leave from vertex \(i\).
3. Let the directed edges \((k−2, k−3), (k−2, k−4), (k−2, k−5)\) and \((k−2, 1)\) all leave from vertex \(k−2\).
4. Let the directed edges \((k−1, k−2), (k−1, k−3), (k−1, k−4), (k−1, 1)\) and \((k−1, 2)\) all leave from vertex \(k−1\).
5. Let the directed edges \((0, k−1), (0, k−2), (0, k−3), (0, 1), (0, 2)\) and \((0, 3)\) all leave from vertex \(0\).

Then, \(\overrightarrow{C}^3(k)\) is persistent.
4. Maintaining Persistence During Certain Formation Changes

In military operations, multi-agent formations can suffer from loss of agents and new agents are required to be added to the (persistent) formations without violating persistence and the existing control structure. Or in some cases, the leader of formation has to be substituted due to evolving mission requirements such as a change in the combat plan. Note that it may be risky to embed all mission plans into one agent (necessarily the leader) since in the case of losing this leader, the formations may malfunction. So it may be practical and safe to have a group of agents carrying different mission plans, and switching mission plans when necessary.

Moreover, as noted in [2] for applications of sensor agents formations, an additional sensor agent may be added to improve the overall coverage, but if the behaviour of each additional sensor is not coordinated with that of its neighbour then considerable time and energy may be wasted and the expected sensing performance will not be realised. Therefore, maintaining persistence as an essential part of coordinated behavior is extremely important.

In this section, we consider all the above practical issues: using the tools allowing transfer of autonomy among agents to assist in change in mission plan/requirement; maintaining persistence while new agent(s) joins formations.

A. Autonomy and Control of 3D Formations

In this subsection, we study the properties of the directed version of Henneberg-like vertex addition [10], which is an abstraction of the event that new agents join a formation, one at each time. Again, we abstract the autonomy of an individual agent as degrees of freedom (DOF) of the vertex representing that agent. We give examples of applying operations to manipulate DOF allocation of persistent graphs, in particular, in three-dimensional spaces.

Let us consider a persistent graph \( G = (V, E) \) in \( \mathbb{R}^d \) \((d \in \{2, 3, \ldots\})\) where \(|V| \geq d\). A directed \(d\)-vertex addition, \( \text{DVA}(d, n) \) where \( n \in \{0, \ldots, d\} \), transforms \( G \) to another persistent graph \( G' = (V', E') \) where \( V' = V \cup \{i\} \), \( E' = E \cup \{(i, k) : \forall k \in V_1 \} \cup \{(j, i) : \forall j \in V_2 \} \), \( V_1 \cup V_2 \subseteq V \), \( V_1 \cap V_2 = \emptyset \), \(|V_1| = d - n \), \(|V_2| = n\), and \( \text{DOF}(j) \geq 1, \forall j \in V_2^1 \) provided that the vertices of \( V_1 \cup V_2 \) do not lie in any \(q\)-dimensional hyperplane where \( q < d\).

We note that from Lemma 2 of [8], constraint consistency is preserved with the directed \(d\)-vertex addition defined above. Moreover, from the following lemma which is drawn from [11], we see that the rigidity is also preserved.

**Lemma 2:** [11] A graph obtained by adding one vertex to a graph \( G = (V, E) \) in \( \mathbb{R}^d \) \((d \in \{2, 3\})\) and \(d\) edges from this vertex to other vertices of \( G \) is rigid if and only if \( G \) is rigid.

Hence by Theorem 2 of [8], the graph obtained after applying a directed \(d\)-vertex addition on a persistent graph

\[ q < d \]

in \( \mathbb{R}^d \) is persistent, i.e., the \(d\)-directed vertex addition defined above preserves the persistence of the graphs.

In the remaining part of this section, we consider in particular 3-dimensional formations. As a more convenient nomenclature in \( \mathbb{R}^3 \), we use the term directed trilateration operation, abbreviated \( \text{DT}(\cdot) \), where \( \text{DT}(n) \equiv \text{DVA}(3, n) \).

An undirected graph formed by a sequence of trilateration operations starting with an initial undirected triangle, often called a trilateration graph, is guaranteed to be generically rigid in \( \mathbb{R}^3 \) and generally globally rigid in \( \mathbb{R}^2 \). A trilateration graph can always be constructed/deconstructed using a polynomial time algorithm, where a reverse trilateration can be performed by removing a vertex with degree 3 at each step. Note that a seed with 3 vertices is needed to initiate a trilateration sequence. However, two different directed triangular seeds can start a directed trilateration operation in \( \mathbb{R}^3 \) as defined in Figure 7(a) and Figure 7(b) are called the leader-first follower-second follower \((L - FF - SF)\) and the balanced triangle \((B_1B_2B_3)\) seeds, respectively.

Specifically in the application to 3-dimensional agent formations, note the meanings of the \( \text{DT}(i) \) operation for different \( i \) can be interpreted as follows:

- \( \text{DT}(3) \) means election of a new leader.
- \( \text{DT}(2) \) may result in either breaking/restoring the balanced control structure, or election of a new first-follower.
- \( \text{DT}(1) \) may also result in either breaking/restoring the balanced control structure at more a detailed level, or creation/change of second follower.
- \( \text{DT}(0) \) preserves the control structure and no decision has to be made by pre-existing agents.

Noting that in a 3-dimensional persistent graph, there are at most 6 DOFs (as opposed to 3 DOFs in the \( \mathbb{R}^2 \) case) to be allocated among the vertices, we can list the following six types of DOF allocation (abbreviated DOF allocation states \( S_1 \)–\( S_6 \) with DOF counts of vertices):

- \( S_1 = \{3, 2, 1, 0, 0, \ldots\} \), \( S_2 = \{2, 2, 2, 0, 0, \ldots\} \)
- \( S_3 = \{3, 1, 1, 1, 0, 0, \ldots\} \), \( S_4 = \{2, 2, 1, 1, 0, 0, \ldots\} \)
- \( S_5 = \{2, 1, 1, 1, 0, 0, \ldots\} \)
- \( S_6 = \{1, 1, 1, 1, 1, 0, 0, \ldots\} \)

Further, we define a transient type of DOF assignment \( S_0 = \{3, 3, 0, 0, \ldots\} \), which can (only) be obtained by applying a \( \text{DT}(3) \) operation to \( S_3 \). \( S_0 \) is named “transient” because it apparently allows two leaders; both have full autonomy to move in 3-dimensional space, simultaneously in control of a formation, and hence this creates an unacceptable situation (termed lack of structural persistence, see [10]) and we want
the DOF assignment to avoid this state. An example of a graph that is in transient state $S_0$ can be seen from Figure 8, if two leaders 1 and 2 go in opposite directions and none of the other three agents is able to maintain desired distances to both.

We study the transformational relationship between the possible distribution of DOFs by applying the appropriate DT($\cdot$) operation using the “state transition diagram” shown in Figure 9. We have the following observations:

- Starting from any one of the two directed triangular seeds, we can build any graph with any of $S_0$–$S_6$ by adding at most three vertices using directed trilateration.
- Any desired DOF reallocation pattern (with no allocation to a specific vertex) can be achieved by at most four directed trilaterations starting with any of the six types of DOF allocation.

The observed results above give an upper bound on the number of agents required to perform a system reconfiguration operation, such as replacement or elimination of the leaders, the first-follower, or the second-follower; or switch to the balanced cooperative control of 3 co-leaders. And it also gives the possible consequences in a closing ranks problem\(^2\), where the lost agent has a certain positive number of DOFs.

5. Concluding Remarks

In this paper, we demonstrated that persistence is a useful concept for formation control. Acquisition of persistence does not come automatically; instead, careful and systematic ways of building persistence are useful. In addition, we demonstrated how to maintain persistence when formations change as when new agents are added. This is particularly useful in formations applied to military operations, where ad-hoc decisions need to be made to control structure of the formation. As complementary studies, we are now working on developing new metrics, to characterize health and robustness of formations; recovering persistence in the event of an agent loss; guaranteeing persistence after merging of two or more persistent formations to accomplish the same mission, as well as testing theoretical results that can be applied to the control of formations of unmanned aerial vehicles.

Acknowledgement

The work of C. Yu, B. Fidan and B.D.O. Anderson is supported by National ICT Australia, which is funded by the Australian Government’s Department of Communications, Information Technology and the Arts and the Australian Research Council through the Backing Australia’s Ability Initiative. Changbin Yu is an Australia-Asia Scholar supported by the Australian Government’s Department of Education, Science and Training through Endeavours Programs. The authors would like to thank Julien M. Hendrickx for his contribution to the theoretical framework on which this paper is built upon.

References


\(^2\)The closing ranks problem for a given rigid formation which has just lost a single agent, is to find new links between some agent pairs which, if maintained cause the resulting formation to again be rigid.