Acquiring and maintaining persistence of autonomous multi-vehicle formations

B. Fidan, C. Yu and B.D.O. Anderson

Abstract: A set of structural cohesiveness issues raised in control of autonomous multi-vehicle formations is analysed, using a recently developed theoretical framework of graph rigidity and persistence. The general characteristics of rigid and persistent formations and some operational criteria to check the rigidity and persistence of a given formation from the aspect of their use in cohesive motion of vehicle formations, including cohesive formation flight is reviewed. Employing these characteristics and criteria, systematic procedures are provided for acquiring and maintaining the persistence of autonomous formations, which are often found in real-world applications. Although these procedures are provided for certain formation classes (in the case of acquisition) or for certain formation operations (in the case of maintenance), the methodology used to develop these procedures has the potential to generate similar procedures for persistence acquisition and maintenance for other formation classes and operations as well.

1 Introduction

Recent research on the general problem of distributed coordinate control of the motion of mobile autonomous vehicle teams, including teams of multiple aerial-vehicles [1–9], has raised a number of issues about acquisition, maintenance and online reconfiguration of vehicle formations. This paper is on analysis and partial solution of such issues, considering the teams of mobile autonomous vehicles as multi-point-agent systems, where the agents can be abstracted as vertices of graphs [10]. The results are applicable to many types of autonomous agent systems including manned or uninhabited aerial vehicles as well as ground vehicles, underwater vehicles, robots and sensor agents.

In this paper, the term formation is used for a collection of agents moving in real two- or three-dimensional space to fulfil certain mission requirements. A formation is called rigid if by explicitly maintaining some distances, the distance between any pair of agents in the formation is consequentially held fixed, and thus the formation can 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 (Fig. 1).

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, for example 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, for example, agent 1 actively maintains its position relative to agent 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. This is modelled in the associated graph by a directed edge from 1 to 2.

In the recent control literature, the characterisation 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], which is called persistence of a directed graph [10] as well. In this paper, we prefer to use the term persistence in order to distinguish it from the undirected and differing graph notion of rigidity.

2 Rigidity and persistence

In this section, we review the notions of rigidity and persistence [10], with a focus on implications of these terms in multi-vehicle formation control applications as well as seeking to provide a theoretical framework for the following sections. Note that although many of the definitions, theorems and other results in this section and the next two can be formulated for arbitrary dimensional space $\Re^n$ ($n \in \{2,3,\ldots\}$) [11], we only consider $n = 2$ or 3, as our focus is on real-world multi-vehicle formation applications.

Acquisition and preservation of structural cohesiveness of an autonomous point-agent formation involves tasks of assigning and maintaining certain distances between nominated pairs of agents. For such tasks, a directed graph $G = (V, E)$, with a vertex set V and a directed edge set E can be used to depict the control architecture as follows [11]: each agent is represented by a vertex in V, and for

doi:10.1049/iet-cta:20050409

Paper first received 15th October 2005 and in revised form 16th June 2006

The authors are with the National ICT Australia Ltd and the Research School of Information Sciences & Engineering, The Australian National University, 216 Northbourne Avenue, Canberra ACT 2601, Australia

E-mail: baris.fidan@nicta.com.au

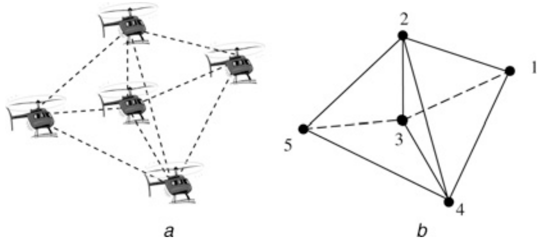

Fig. 1 *Aerial-vehicle formation*

a Three-dimensional rigid aerial-vehicle formation
b Underlying undirected graph
There is no active attempt to hold the distance between 1 and 5 fixed; rigidity ensures this

each agent (vertex) pair $i, j \in V$ there is a directed edge $(\vec{i,j})$from i to j, if i has a constraint on the distance it must actively maintain from j. We call G the underlying directed graph of the formation. The undirected graph with the same vertex set V and the same edge set E but with the edge directions neglected is called the underlying undirected graph of the formation (or of G).

In the recent control literature, the characterisation of a point-agent formation of the above type has started to be attempted using the notion of rigidity of a directed graph [1], which is called persistence of a directed graph [10] as well. In this paper, we prefer to use the term persistence in order to distinguish it from the undirected graph notion of rigidity.

2.1 Rigidity

We define rigidity of a formation formally in the following paragraphs. Intuitively, a formation (which can be represented by a graph in which vertices have locations in $\Re^2$ or $\Re^3$) is rigid if by explicitly maintaining some distances between certain agent pairs of the formation (the pairs being joined by an edge in the underlying graph) distances between all agent pairs are consequentially maintained. It turns out that the question of whether or not a formation is rigid can for generic formations be settled just by examining the underlying graph. We now express these ideas more formally.

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

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

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

A representation p is rigid if there exists $\epsilon > 0$ such that for all realisations p' of the distance set induced by p and satisfying $\delta(p, p') < \epsilon$, 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 previous works [10, 12]. One reason for using these terms, in $\Re^n$ $(n \in \{2,3\})$, is to avoid the problems arising from having $n+1$ or more vertices lying in a n_1-dimensional space where $n_1 \leq n - 1$.

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. Fundamental characteristics of rigid and minimally rigid graphs and some of their applications in autonomous formation control can be found in some previous works [5, 12–15]. We conclude this subsection with a lemma on one of these characteristics, which follows from the Henneberg construction principle [12–14] and is used in the subsequent sections.

Lemma 1: A graph obtained by adding one vertex to a graph $G = (V, E)$ in $\Re^n$ $(n \in \{2, 3\})$ and n edges from this vertex to other vertices of G is (minimally) rigid if and only if G is (minimally) rigid.

2.2 Persistence

Persistence has the following intuitive meaning, valid for almost all formations in $\Re^2$ and $\Re^3$. A formation is persistent if, provided that all the agents are trying to satisfy the distance constraints on them, they can, in fact, satisfy these constraints and, consequently, the global structure of the formation is preserved, that is when the formation moves, it necessarily moves as a cohesive whole (There exists an exceptional small class of formations in $\Re^3$, for which the intuitive explanation here and the formal definition of persistence given in Section 2.2 do not match. This class is further discussed in Section 2.3.). In order to have a formal definition of persistence, we first need to characterise the fact that each agent is trying to keep the distances from its neighbours constant.

Let us thus fix a directed graph G, desired distances $\delta_{ij} > 0$ for $\forall\ (i, j) \in E$, and a representation p. We say that the edge $(i, j) \in E$ is active if $||p(i) - p(j) = \delta_{ij}$. Given a representation p, We also say that the position of the vertex $i \in V$ is fitting for the distance set $\bar{d}$ 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 $p^* \in \Re^n$ for which the following strict inclusion holds

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

This condition intuitively means that the agent i cannot satisfy additional distance constraints without breaking some that it already satisfies.

A representation of a graph is a fitting representation for a certain distance set $\bar{\delta}$ if all the vertices are at fitting positions for $\bar{\delta}$. Note that any realisation 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 $\delta(p, p') < \epsilon$ is congruent to p. A graph is then generically persistent if almost all its representations are persistent.

As stated in Theorem 1, a generically persistent graph in $\Re^n$ $(n \in \{2, 3\})$ is always generically rigid, and there exists 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 $\bar{\delta}$ induced by p and satisfying $\delta(p, p') < \epsilon$ is a realisation of $\bar{\delta}$. 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 $\Re^2$ can be found in the work of Hendrickx *et al.* [10]. Again, we say that a graph is generically constraint consistent if almost all its representations are constraint consistent. The relation among persistence, rigidity and constraint consistence is given in the following theorem and demonstrated using a two-dimensional example in Fig. 2.

Theorem 1: A representation in $\Re^n$ ($n \in \{2, 3\}$ 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 [11].

Before discussing some practical implications of the notions of rigidity and persistence, in order to complete our theoretical framework needed to establish the results in the later sections, we present some fundamental results on persistence characterisation. A complete list of the basic properties of persistent formations and their detailed analysis can be found in the work of Hendrickx and co-workers [10, 11].

In order to check persistence of a directed graph G, one may use the following criterion, where $d^-(i)$ and $d^+(i)$ designate, respectively, the in- and out-degree of the vertex i in the graph G, viz. the number of edges in G heading to and originating from i, respectively.

Proposition 1: A persistent graph in $\Re^n$ ($n \in \{2, 3\}$) remains persistent after deletion of any edge $(\vec{i,j})$ for which $d^+(i) \geq n + 1$. Similarly, a constraint consistent graph in $\Re^n$ ($n \in \{2, 3\}$) remains constraint consistent after deletion of any edge $(\vec{i,j})$ for which $d^+(i) \geq n + 1$ [11].

A useful concept in analysing acquisition and maintenance of persistence is the number of degrees of freedom (DOF count) of a vertex, which is defined as the maximal dimension, over all representations of the graph, of the set of possible fitting positions for this vertex. For example, in $\Re^2$, the vertices with zero out-degrees have two DOFs, the vertices with out-degrees 1 have one DOF and the others have zero DOF. An n-DOF vertex (agent) in an n-dimensional graph (formation) is called a leader. Note that a vertex with zero DOF can have more than one possible fitting position. Observe indeed that, in almost all situations in $\Re^n$ ($n \in \{2, 3\}$), there are two possible fitting positions for a vertex with out-degree n. However, since

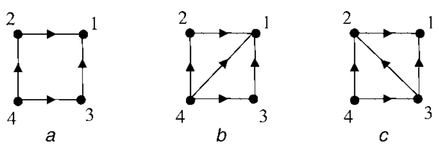

Fig. 2 *Application of theorem 1 in $\Re_2$*

a The graph is constraint consistent but not rigid, hence it is not persistent
b The graph is rigid but not constraint consistent [observe that vertices (agents) 1, 2 and 3 can move to new positions, without violating the distance constraints on them, for which vertex 4 is unable to meet all the three distance constraints on it at the same time.], hence it is not persistent
c The graph is both rigid and constraint consistent, hence it is persistent

this set contains a finite number of points, its dimension is still 0. The following immediate corollary of Proposition 1 provides a natural bound on the total number of DOFs in a persistent graph, which we also call the total DOF count of that graph.

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

We have stated in Proposition 1 that a persistent graph remains persistent after deletion of any edge (i, j) for which $d^+(i) \geq n + 1$. 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 n. The next theorem, which is a special case of Theorem 9 in the work of Yu *et al.* [11], states that a graph is persistent if and only if all the graphs obtained in this way are rigid in $\Re^2$.

Theorem 2: An n-dimensional ($n \in \{2, 3\}$) 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 n until all such vertices have an out-degree equal to n.

Theorem 2 provides a non-polynomial time algorithm to check the persistence of any n-dimensional graph for $n \in \{2, 3\}$: 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 $n + 1$ until all the vertices have an out-degree less or equal to n. 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.

We conclude our review on the characteristics of persistent formations with the following proposition [11] and is used in the subsequent sections.

Proposition 2: Consider a directed graph G in $\Re^n$ ($n \in \{2, 3\}$) and another graph G' that is obtained by adding one vertex to G and at least n edges leaving this vertex. Then, G' is persistent if and only if G is persistent [11].

2.3 Structural persistence

It has been mentioned in the beginning of Section 2 that persistence, for almost all formations in $\Re^2$ and $\Re^3$, means ability of the agents to satisfy the distance constraints on them and, as a consequence, preservation of the global formation structure and cohesive motion, provided that the agents are trying to satisfy their constraints. In $\Re^3$, however, there exists a small class of formations that are persistent by the formal definition given in Section 2.2, but can still have problems in moving as a cohesive whole [11]. The problem in this exceptional class arises when it is not possible to satisfy all the constraints on all the agents in the formation at the same time, despite ability of any single agent to move to a position which satisfies the constraints on it once all the other agents are fixed.

Persistent graphs free of the above problem are called structurally persistent [11]. For formal definition and characteristics of structural persistence as well as the details of the above problem, the reader may refer to the work of Yu *et al.* [11]. In effect, structural persistence is equivalent to requiring all subsets of agents to be fitting, as opposed to requiring all single agent subsets being fitting (for persistence). The distinction between structural persistence and persistence does not arise in two-dimensions. In $\Re^3$, it turns out that a formation is structurally persistent if and only if it is persistent and does not have two leaders each with 3 DOFs. This case, as a situation to be avoided during formation changes, is briefly discussed in Section 4. Other than this brief discussion, throughout the paper, we ignore the small class of three-dimensional formations which are persistent but not structurally persistent, that is all the practical persistent formations in the paper are assumed to be structurally persistent as well.

2.4 Real-world implications of persistence

Persistence (or structural persistence in three-dimensions) is the crucial property of an information/control architecture that ensures that a formation exhibits cohesive behaviour. Minimal persistence defines those situations where loss of a link will destroy cohesiveness, and from an operational point of view, non-minimal persistence may be desirable to secure redundancy. Indeed, one can lose any $d^+(i) - n$ outward links from vertex i (if $d^+(i) > n$) and retain persistence. The persistence concept is needed to study formations with leader–follower structure [1, 2, 4, 6, 8, 9]; such structures have for some time been postulated as appropriate, but without a full analytical framework for analysing them.

Note here that in this paper, since our focus is acquisition and maintenance of persistence from the overall formation perspective, we do not consider the effects of dynamics of each individual agent or their interactions on the overall formation behaviour. Nevertheless, agent dynamics and dynamic interactions are major issues in real-world multi-vehicle formation control and some further discussions on these issues can be found in the work of Sandeep *et al.* [16] and the references therein.

In the implementation of the persistence acquisition and maintenance strategies presented in this paper, which is partially discussed in a previous work [15], a common requirement would be developing decentralised controllers for individual agents, instead of a centralised control scheme. The main concerns leading to this requirement are complexity and computational cost, sensitivity to loss of certain agents, (e.g. a central commander), communication delays between the commander agent and the other agents, impracticality of processing local information by a central control unit and so on in a possible central control scheme.

Having provided a theoretical framework of rigidity and persistence and some practical implications of this framework in the real-world, in the next section, we turn our attention to the problem of systematic construction of provably persistent two-dimensional formations by assigning directions to the links.

3 Persistence acquisition for autonomous formations

In this section, we study how formations with special classes of underlying rigid graphs can acquire persistence systematically. Note that a formation based on a persistent graph offers often a reduction in the control complexity by half as compared to the one based on an (undirected) rigid graph, 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 two-dimensional formations as the underlying rigid graphs, and each directed edge $(\overrightarrow{i,j})$ corresponds to agent i actively maintaining its distance from agent j.

In Section 3.1, we consider certain classes of rigid undirected graphs and seek procedures to assign directions to the edges of these undirected graphs in order to obtain persistent directed graphs. In Section 3.2, we discuss the practical applications of these procedures in acquiring certain classes of autonomous formations.

3.1 Persistence acquisition for certain rigid graphs

Direction assignment to an arbitrary rigid undirected graph in order to obtain a persistent directed graph is currently an open problem, as can be expected from the facts that the notion of persistence is very recently defined and the relation between this directed graph notion and the undirected graph notion of rigidity is non-trivial. Here, we consider certain classes of rigid graphs; namely complete graphs, bilateration and trilateration graphs, wheel graphs, C^2 graphs, C^3 graphs and bipartite graphs of type $K_{m,n}$ [17, 18]. The practical formations corresponding to these classes are described in Section 3.2.

For each of these rigid graph classes, we provide a procedure of assigning directions to the edges for acquiring persistence of the resultant directed graph, each summarised as a separate proposition. The formal definition of each class is embedded in the corresponding proposition. Rigidity of each class can be easily verified using the definitions and rigidity criteria available in the literature [12–14].

Proposition 3: Given an integer $k \geq 3$, consider the k-complete (undirected) graph K_k in $\Re^n$ ($n \in \{2, 3\}$) with the vertex set $V = \{1, 2, \ldots, k\}$, where every vertex pair $i, j \in V$ is directly connected by an edge. Let $\bar{K}_k$ be the directed graph obtained by assigning directions to the edges of K_k such that for any vertex pair i, j satisfying $1 \leq i \leq j \leq k$, direction of edge (i, j) is from j to i. Then, $\bar{K}_k$ is persistent (Fig. 3).

Proof: It is easy to verify that $\bar{K}_3$ is persistent. To show that $\bar{K}_k$ is persistent for any k, we use induction, that is we prove that $\bar{K}_{k+1}$ is persistent provided that $\bar{K}_k$ is persistent.

Now, assume that $\bar{K}_k$ is persistent and consider any subgraph $\bar{G}_k$ of $\bar{K}_k$ that is obtained by removing outgoing edges from vertices with out-degree larger than n until every vertex has an out-degree less than or equal to n. According to Theorem 2, $\bar{G}_k$ is rigid.

On the other hand, $\bar{K}_{k+1}$ consists of $\bar{K}_k$, vertex $k+1$ and edges $(\overrightarrow{k+1, i})$ for $i = 1, 2, \ldots, k$, each of which is directed from vertex $k+1$ to vertex i. Hence, defining the subgraphs $\bar{G}_{k+1}$ of $\bar{K}_{k+1}$ similar to $\bar{G}_k$s, any G_{k+1} consists of a G_k, vertex $k+1$ and n edges connecting vertex $k+1$ to $\bar{G}_k$. Therefore since any $\bar{G}_k$ is rigid, because of Lemma 1, any $\bar{G}_{k+1}$ is rigid as well, which in turn implies due to Theorem 2 that $\bar{K}_{k+1}$ is persistent. □

Proposition 4: Given a bilateration graph T in $\Re^2$, viz. a graph with an ordering of vertices $1, 2, \ldots, k$ such that 1, 2 and 3 form a complete graph, and vertex j is joined to at least two of vertices $1, 2, \ldots, j-1$ for $j = 4, 5, \ldots, k$, let $\bar{T}$ be the directed graph obtained by assigning directions to the edges of T such that direction of each edge (i, j) for $i \leq j$ is from j to i. Then, $\bar{T}$ is persistent (Fig. 4).

Proof: The complete graph with vertices 1, 2, 3 is clearly persistent after direction assignment. The directed graph $\bar{T}$ formulated in the proposition can be constructed starting with this complete graph and progressively for $j = 4, 5, \ldots, k$ adding vertex j and the edges of T between j and $\{1, 2, \ldots, j-1\}$ each with a direction leaving from j. According to Proposition 2, the directed graph formed at each step of this procedure and therefore $\bar{T}$ is persistent. □

hence $C^2(k)$ is rigid as well. Furthermore each of the vertices of $\bar{C}^2(k)$ has an out-degree 2. Therefore the result follows as a direct consequence of Theorem 2. □

Proposition 8: Given an integer $k \geq 3$, consider the graph $C^3(k)$ in $\Re^n$ ($n \in \{2, 3\}$) that is composed of vertices 0, 1, $\ldots$, $k-1$ and edges $(i, (i+j) \bmod k)$ for $i = 0, 2, \ldots, k-1$ and $j = 1, 2, 3$. Let $\bar{C}^3(k)$ be the directed graph obtained by assigning directions to the edges of $C^3(k)$ using the following procedure (Fig. 7).

(1) For the subgraph of $C^3(k)$ that is composed of vertices 1, 2, 3, 4 and the edges among them, which is a complete graph, assign directions according to Proposition 3.
(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 the three directed edges leave from vertex i.
(3) Let the directions of the edges $(k-2, k-3)$, $(k-2, k-4)$, $(k-2, k-5)$ and $(k-2, 1)$, after direction assignment, all leave from vertex $k-2$.
(4) Let the directions of the 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 directions of the edges $(0, k-1)$, $(0, k-2)$, $(0, k-3)$, $(0, 1)$, $(0, 2)$ and $(0, 3)$ all leave from vertex 0.

Then, $\bar{C}^3(k)$ is persistent.

Proof: The directed complete graph with four vertices in step (1) is persistent due to Proposition 3. $\bar{C}^3(k)$ can be formed starting with this complete persistent graph and progressively for $j = 5, 6, \ldots, k-2, k-1, 0$ adding vertex j and the directed edges mentioned in steps (2)–(5) that are leaving vertex j. According to Proposition 2, the directed graph formed at each step of this procedure and therefore $\bar{C}^3(k)$ is persistent. □

Proposition 9: Given two integers $m \geq 3$ and $k \geq 3$ consider the bipartite graph $K_{m,k}$ in $\Re^2$ that is composed of $m+k$ vertices $v_{1,1}$, $v_{1,2}$, $\ldots$, $v_{1,m}$, $v_{2,1}$, $v_{2,2}$, $\ldots$, $v_{1,k}$ and mk edges $e_{i,j} \stackrel{\Delta}{=} (v_{1,i}, v_{2,j})$ for $i = 1, 2, \ldots, k$ and $j = 1, 2, \ldots, k$. Let $\bar{K}_{m,k}$ be the directed graph obtained by assigning directions to the edges of $K_{m,k}$ using the following procedure (Fig. 8).

(1) For $i = 1, 2, 3$ and $j = 1, 2, 3$, let the direction of edge $e_{i,j}$ be from $v_{1,i}$ to $v_{2,j}$ if $i = j$ and from $v_{2,j}$ to $v_{1,i}$ if $i \neq j$.
(2) (if $m \geq 4$) For $i = 4, 5, \ldots, m$, assign the directions of the edges $e_{i,1}$, $e_{i,2}$ and $e_{i,3}$ such that all the three directed edges leave from vertex $v_{1,i}$.
(3) (if $k \geq 4$) For $i = 1, 2, \ldots, m$ and $j = 4, 5, \ldots, k$, let the directed edges $e_{i,j}$ all leave from vertex $v_{2,j}$.

Then, $\bar{K}_{m,k}$ is persistent.

Proof: Rigidity of $K_{3,3}$ can be easily shown using the techniques in the work of Tay and Whiteley [12] and Whiteley [14]. Hence, the directed graph with six vertices and nine edges in step (1) is persistent due to Theorem 2. $\bar{K}_{m,k}$ can be formed starting with this complete persistent graph, progressively for $i = 4, 5, \ldots, m$ adding vertex $v_{1,i}$ and the directed edges mentioned in step (2) that are leaving vertex $v_{1,i}$, and then for $i = 1, 2, \ldots, m$ adding vertex $v_{2,j}$ and the directed edges mentioned in step (3) that are leaving vertex $v_{2,j}$. According to Proposition 2, the directed graph formed at each step of this procedure and therefore $\bar{K}_{m,k}$ is persistent. □

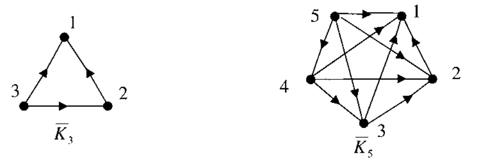

Fig. 3 *Direction assignment to complete underlying graphs*
Persistence acquisition of formations where each agent is neighbour of any other agent

Proposition 5: Given a trilateration graph T in $\Re^n$ ($n \in \{2, 3\}$), viz. a graph with an ordering of vertices $1, 2, \ldots, k$ such that 1, 2 and 3 form a complete graph, and vertex j is joined to at least three of vertices $1, 2, \ldots, j-1$ for $j = 4, 5, \ldots, k$, let $\bar{T}$ be the directed graph obtained by assigning directions to the edges of T such that direction of each edge (i, j) for $i < j$ is from j to i. Then, $\bar{T}$ is persistent.

Proof: The proof can be obtained applying exactly the same steps of the proof of Proposition 4. □

Remark 1: Any directed k-vertex bilateration or trilaterion graph $\bar{T}$ described in Propositions 4 and 5 can be obtained from the directed complete graph $\bar{K}_k$ described in Proposition 3, by obeying the same vertex ordering and removing some of the edges.

Proposition 6: Given an integer $k \geq 3$, consider the wheel graph W_k in $\Re^2$ 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 $\bar{W}_k$ 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, $\bar{W}_k$ is persistent (Fig. 5).

Proof: Since none of the vertices of $\bar{W}_k$ has an out-degree larger than 2, the result directly follows from Theorem 2. □

Proposition 7: Given an integer $k \geq 3$, consider the graph $C^2(k)$ in $\Re^2$ that is composed of vertices $0, 1, \ldots, k-1$ and edges $(i, (i+j) \bmod k)$ for $i = 0, 2, \ldots, k-1$ and $j = 1, 2$. Let $\bar{C}^2(k)$ be the directed graph obtained by assigning directions to the edges of $C^2(k)$ such that direction of each edge $(i, (i+j) \bmod k)$ is from i to $(i+j) \bmod k$. Then, $\bar{C}^2(k)$ is persistent (Fig. 6).

Proof: Since the subgraph of $C^2(k)$ that is obtained by removing edges $(0, k-1)$, $(1, k-1)$, $(0, k-2)$ can also be obtained via a Henneberg sequence [12], that is starting with the K_3 graph with vertices 0, 1, 2 and progressively for $i = 3, 4, \ldots, k-1$ adding vertex i and edges $(i, i-1)$ and $(i, i-2)$. Because of Lemma 1, this subgraph is rigid and

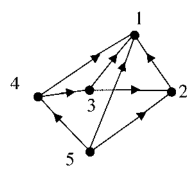

Fig. 4 *Direction assignment to a bilateration underlying graph*
Persistence acquisition of a two-dimensional formation in leader–follower structure

Proposition 10: Given two integers m, k satisfying $k \geq m \geq 4$ and $k + m \geq 10$, consider the bipartite graph $K_{m,k}$ in $\Re^3$ that is composed of $m + k$ vertices $v_{1,1}$, $v_{1,2}$, $\ldots$, $v_{1,m}$, $v_{2,1}$, $v_{2,2}$, $\ldots$, $v_{1,k}$ and mk edges $e_{i,j} \triangleq (v_{1,i}, v_{2,j})$ for $i = 1, 2, \ldots, m$ and $j = 1, 2, \ldots, k$. Let $\bar{K}_{m,k}$ be the directed graph obtained by assigning directions to the edges of $K_{m,k}$ using the following procedure.

(1) For $i = 1, 2, 3, 4$ and $j = 1, 2, 3, 4, 5$, let the direction of edge $e_{i,j}$ be from $v_{1,i}$ to $v_{2,j}$ if $i = j$ or $i = j - 1$, and from $v_{2,j}$ to $v_{1,i}$ otherwise.
(2) (if $k \geq 6$) For $i = 1, 2, 3, 4$, let the directed edge $e_{i,6}$ leave from vertex $v_{1,i}$.
(3) (if $k \geq 7$) For $i = 7, 8, \ldots, k$, assign the directions of the edges $e_{1,i}$, $e_{2,i}$, $e_{3,i}$ and $e_{4,i}$ such that all the four directed edges leave from vertex $v_{2,i}$.
(4) (if $m \geq 5$) Let the directions of the edges $e_{5,2}$, $e_{5,3}$, $e_{5,4}$ be all towards $v_{1,5}$. Assign the directions of $e_{5,1}$ and $e_{5,5}$, $\ldots$, $e_{5,k}$ to all leave from $v_{1,5}$.
(5) (if $m \geq 6$) For $i = 6, 7, \ldots, m$ and $j = 1, 2, \ldots, k$, let the directed edges $e_{i,j}$ all leave from vertex $v_{1,i}$.

Then, $\bar{K}_{m,k}$ is persistent.

Proof: It is shown that any three-dimensional bipartite graph $K_{m,k}$, where $k \geq m \geq 4$ and $k + m \geq 10$, is rigid [14]. Using this fact together with Theorem 2, it can be easily verified that the directed graphs $\bar{K}_{4,6}$ [obtained applying steps (1) and (2)] and $\bar{K}_{5,5}$ [obtained applying steps (1) and (4)] are both persistent. Furthermore, any $K_{m,k}$ with $k \geq m \geq 4$ and $k + m \geq 10$ has to contain $K_{4,6}$ unless $m = k = 5$, that is $\bar{K}_{m,k} = \bar{K}_{5,5}$, which is persistent. Then, for $\bar{K}_{m,k} \neq \bar{K}_{5,5}$, since $\bar{K}_{4,6}$ is persistent, and steps (3)–(5) preserve persistence because of Proposition 2 and Theorem 2, the resultant $\bar{K}_{m,k}$ is also persistent. □

3.2 Applications of persistence acquisition

Each of the rigid graph classes considered in Section 3.1 corresponds to a formation architecture that can be used in guidance and control of aerial-vehicle teams. Complete graphs model the concept of closed neighbourhood of vehicles 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 neighbourhood. Note that the persistence acquisition procedure described in Proposition 3 is valid in both $\Re^2$ and $\Re^3$. Hence such closed neighbourhoods can be acquired for three-dimensional formations as well as the two-dimensional ones.

Bilateration and trilateration results given in Propositions 4 and 5 can be used in acquisition of cycle-free formations with leader–follower structure [9] and asymmetric control/communication architecture. More on persistence of cycle-free formations will be covered in Section 4.

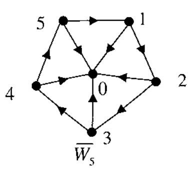

Fig. 5 *Direction assignment to a wheel graph*

Persistence acquisition of a two-dimensional formation with a central commander

One might use wheel graphs, the persistence acquisition procedure for which is given in Proposition 6, to model two-dimensional formations of flying vehicles, in which there is a central vehicle, the commander, which can be sensed by all other vehicles. It is worth mentioning that the sensing/communication capabilities of the vehicles and hence the distance constraints on them may be different from each other. Hence the formations of this type may have various geometries, not necessarily having the commander in the geometric centre as in the case depicted in Fig. 5.

The C^2 and C^3 graphs described in Propositions 7 and 8 illustrate the scenarios when the sensing/communication/control region of each vehicle in a cyclic formation is doubled and tripled, respectively, so that extra links between agents are established. The graphs provide some measure of redundancy in that one or more links or even an agent with its incident links can be lost with persistence being maintained. The wheel graph discussed earlier has a similar redundancy property as well. Though in contrast to the C^2 and C^3 graphs, there is a distinguished agent, the centrally located agent or the 'commander', the loss of which would destroy the persistence, indeed rigidity, of the formation (Figs. 6 and 7).

Bipartite graphs, in general, abstract formations that contain two types of vehicles (with different functionalities) where each vehicle interacts only with the vehicles of the other type. One potential application area of such an abstraction is surveillance tasks involving a formation with two sets of vehicles S_1 and S_2, where each vehicle in S_1 is responsible for communicating with a base station to generate high level commands to be sent to the vehicles in S_2, independently from the other vehicles in S_1, and each vehicle in S_2 is responsible to perform an individual (low level) local surveillance task according to the commands it receives from each of the vehicles in S_1. Acquisition and maintenance of the control structure between S_1 and S_2, which can be represented by a bipartite graph is essential in order to perform the general surveillance task properly (Fig. 8).

Although the procedures in this section are provided for a limited number of formation classes, the methodology used to develop these procedures can be used to generate similar

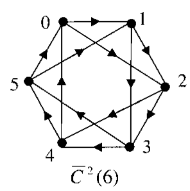

Fig. 6 *Direction assignment to a C^2 graph*

Persistence acquisition of a formation with doubled communication radius

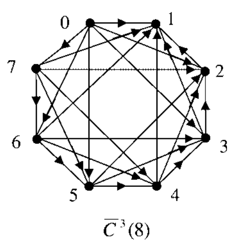

Fig. 7 *Direction assignment to a C^3 graph*

Persistence acquisition of a formation with tripled communication radius

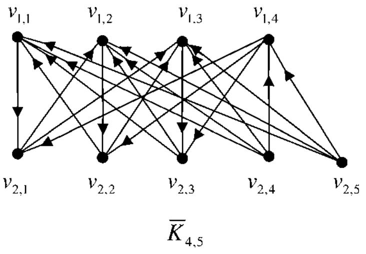

Fig. 8 *Direction assignment to a bipartite graph*

procedures for persistence acquisition and maintenance of other formation classes as well.

4 Maintaining persistence during formation changes

In many multi-vehicle formation applications such as the ones involving military surveillance operations, as a matter of guaranteeing robustness, one needs to consider certain scenarios that have a significant probability to happen. One such scenario is where a multi-agent (vehicle) formation loses some of its agents and new agents are required to be added to the formation without violating the existing control structure [5]. If this formation is persistent in the beginning in terms of its control and sensing structure, maintenance of the control structure during the formation change can be abstracted as maintaining the persistence.

A similar scenario is where the leader of a formation has to be substituted due to evolving mission requirements such as a change in the combat plan. Note here that it may be risky to embed all mission plans into a single agent, that is the leader, since in the case of losing this leader, the formation may malfunction. So it may be practical and safe to use a group of agents to carry different mission plans and switch/transfer these plans when necessary. If the formation in the beginning is persistent, the leader-change task mentioned above can be abstracted without damaging the control structure by changing the direction of certain edges in the underlying directed graph in an appropriate way that maintains the persistence.

Another application where frequent formation changes are expected is surveillance of regions of interest using a formation of sensors mounted on aerial vehicles [3]. Abstracting each vehicle with the sensing/communication equipment on it as a sensor agent, in order to adapt varying conditions during the surveillance mission, an extra sensor agent may be needed to improve the overall coverage. If the behaviour of each such additional sensor is not coordinated well with that of its neighbours, then considerable time and energy may be wasted and the expected sensing performance may not be achieved. In order to prevent such a case, it is important to maintain persistence of the formation during variations.

In this section, to address some structural issues that may appear during formation changes, examples of which are given above, we elaborate on the maintenance of persistence in autonomous formations during such changes, using some graph theoretical tools allowing transfer of degree of autonomy among agents. We use a recently introduced directed version of Henneberg-like vertex addition [11] to abstract joining of new agents in a formation, one at each time. We abstract the autonomy of an individual agent as DOF of the vertex representing that agent. On the basis of this abstraction, we give examples of operations that manipulate DOF allocation of persistent graphs, in particular, in three-dimensions, corresponding to degree of autonomy allocation among agents of a formation.

Note that in a three-dimensional persistent graph (with $n = 3$), there are at most $n(n+1)/2$=6 DOFs, as opposed to three DOFs in the $\Re^2$ case, to be allocated among the vertices [11]. This allocation can be performed in six different ways (considering the agents as indistinguishable) which can be represented by the following six DOF allocation states, that is sets of DOF counts of vertices ordered in a non-increasing manner

$$S_1 = \{3, 2, 1, 0, 0, \ldots\}, \quad S_2 = \{2, 2, 2, 0, 0, \ldots\},$$
$$S_3 = \{3, 1, 1, 1, 0, 0, \ldots\} \quad S_4 = \{2, 2, 1, 1, 0, 0, \ldots\},$$
$$S_5 = \{2, 1, 1, 1, 1, 0, 0, \ldots\}, \quad S_6 = \{1, 1, 1, 1, 1, 1, 0, 0, \ldots\}$$

Thus, in S_4, two agents have two DOFs and two agents have one DOF. In the above list, we have skipped a last possible DOF allocation state, $S_0 = \{3, 3, 0, 0, \ldots\}$, since it is an undesired one that can be accepted only temporarily during formation changes as a transient state. This is because S_0 allows two leaders that both have full autonomy to move in $\Re^3$, simultaneously in control of a formation, generating an unacceptable situation of lack of structural persistence, which has been briefly discussed in Section 2.3. An example of a formation whose underlying directed graph is in the transient state S_0 is depicted in Fig. 9; observe that if the two leaders 1 and 5 go in opposite directions, then the other three agents will not be able to maintain all their desired distances.

Next we focus on maintaining persistence of a three-dimensional formation during addition of a new agent to the formation and perform and analyse based on the DOF allocation states defined above. In order to form a graph theoretical framework for the operation of adding a new agent to the formation, we first define a set of directed vertex addition operations requiring minimal number of new edges for maintaining persistence.

Consider a persistent graph $G = (V, E)$ in $\Re^3$ where $|V| \geq 3$. A directed trilateration [11], DT(m) where $m \in \{0, 1, 2, 3\}$, is a transformation of G to another persistent graph $G' = (V', E')$ where $V' = V \cup \{i\}$, $E' = E \cup \{(\vec{i,k})$: $\forall k \in V_1\} \cup \{(\vec{j,i})$: $\forall j \in V_2\}$ for some $V_1, V_2 \subseteq V$ satisfying $V_1 \cap V_2 = \emptyset$, $|V_1| = 3 - m$, $|V_2| = m$ and $DOF(j) \geq 1$, $\forall j \in V_2$, provided that the vertices of $V_1 \cup V_2$ are all distinct and are not collinear (If there is no such V_2 then the corresponding DT(m) cannot be performed for the graph G.).

Remark 2: Using an argument based on Theorem 2, it can be shown that the graph obtained after applying a directed trilateration in $\Re^3$ is persistent, that is, the directed trilateration defined above preserves the persistence of the graphs.

Remark 3: An undirected graph formed by applying a sequence of trilateration operations starting with an initial undirected triangle, often called a trilateration graph, is

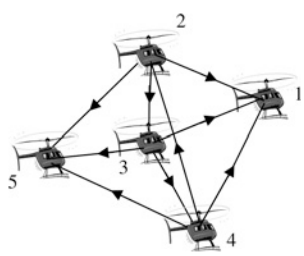

Fig. 9 *Transient DOF allocation state, S_0*

Three-dimensional aerial-vehicle formation with two leaders (1 and 5)

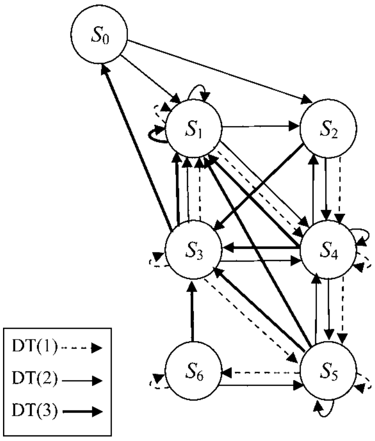

Fig. 10 *DOF allocation state transition diagram for directed trilaterations*

guaranteed to be generically rigid in $\Re^3$ [13, 14]. Similarly, Remark 2 implies that a directed graph formed by applying a sequence of directed trilateration operations starting with any initial directed triangle with three vertices and three directed edges, one for each vertex pair is guaranteed to be generically persistent in $\Re^3$, since any such initial directed triangle is persistent.

The four directed trilateration operations DT(0), DT(1), DT(2), DT(3) can be considered as transition operations to change the DOF allocation state of a formation, that is to transit from one of the states (S_0,) $S_1, \ldots, S_6$ to another. Note that DT(0) operations do not change the DOF allocation state of a formation, since they do not change the DOF counts of the existing vertices and add a zero DOF agent to the formation. All the possible DOF allocation state transitions using directed trilateration are summarised in a diagram in Fig. 10.

From Fig. 10, one can observe that the transient DOF allocation state S_0 can be reached in a single step only from S_3, applying a DT(3) operation. Hence, one needs to be careful at state S_3 in order to avoid reaching S_0, that is a situation of lack of structural persistence that can be caused by addition of a new agent (vertex) to the formation (graph). It can also be verified using Fig. 10 that starting from any directed triangle described in Remark 3, we can build a graph (formation) in any of the DOF allocation states S_0–S_6 by adding at most three vertices (agents) using directed trilateration. Fig. 10 further implies that any desired DOF reallocation pattern (with no allocation to a specific vertex) can be achieved by at most four directed trilaterations starting at any of the six DOF allocation states. An example for persistence maintaining formation changes in the light of the last two observations above is depicted in Fig. 11, where a formation in DOF allocation state S_2 is brought to state S_1 using two consecutive directed trilateration operations.

The above observations can be used to deduce the following interpretations of the directed trilateration operation DT(m) for different values of m: DT(0) preserves the control structure and no decision has to be made by pre-existing agents; the new agent follows three of the pre-existing agents. DT(3) means assignment of the new agent as the (new) leader. If the formation initially has a leader, that is at state S_1 or S_3, DT(1) and DT(2) may either make the formation leader-less or may result in reallocation of DOFs without changing the leader. If the formation initially does not have any leader then both DT(1) and DT(2) reallocate DOFs without changing the leader-less structure.

The observations on Fig. 10 also provide an upper bound on the number of agents required (to be added) to perform a system reconfiguration operation, such as replacement of the leaders, switching between a leader–follower structure and a balanced (leader-less) structure. Considering inverse transitions, the same observations may be applied to the analysis of persistence maintenance during a closing ranks [5] scenario, that is during establishing new directed links between existing agent pairs of a persistent formation after losing a single agent having a certain positive DOF count in order to maintain persistence.

5 Concluding remarks

In this paper, we have demonstrated that persistence is a useful concept in the control of multi-vehicle formations. We have reviewed the general characteristics of persistent formations and a set of operational criteria to check/guarantee persistence in order to build a theoretical framework to be used in designing control schemes for cohesive motion of vehicle formations. We have developed systematic procedures for acquiring the persistence of some classes of autonomous formations, which are often found in real-world applications. Although these procedures are provided for a limited number of formation classes, the methodology used to develop these procedures has the potential to generate similar procedures for persistence acquisition of other formation classes as well.

Later, we have analysed maintenance of persistence during formation changes, particularly during addition of new agents. This is particularly useful in formation tasks where *ad hoc* decisions need to be made to control structure of the formation, such as multi-vehicle surveillance operations over unsteady terrains. As complementary studies, we currently work on developing new metrics to characterise 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

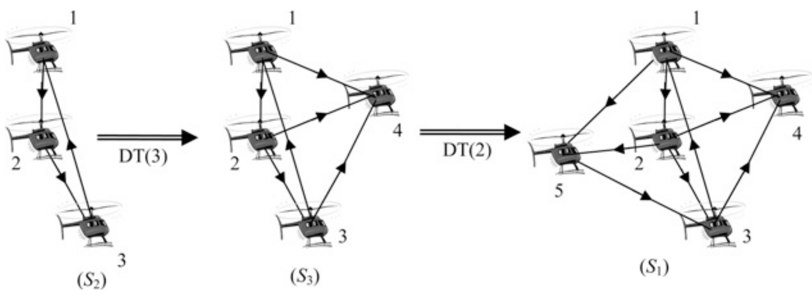

Fig. 11 *State transition from S_2 to S_1 via a pair of consecutive directed trilateration operations*

Leader-less 3-agent formation is changed into a 5-agent formation with a leader–follower structure results that can be applied to the control of formations of aerial vehicles.

6 Acknowledgments

This work 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. C.Yu. is an Australia–Asia Scholar supported by the Australian Government's Department of Education, Science and Training through Endeavours Programs. A subset of the ideas of this paper in a more condensed form was presented in the 2nd International Conference on Intelligent Sensors, Sensor Networks and Information Processing, Melbourne, Australia, December 2005 under the title 'Persistence acquisition and maintenance for autonomous formations'. The authors would like to thank Julien M. Hendrickx and Vincent D. Blondel for their contribution to the theoretical framework on which this paper is built upon.

7 References

1 Baillieul, J., and Suri, A.: 'Information patterns and hedging Brockett's theorem in controlling vehicle formations'. Proc. 42nd IEEE Conf. on Decision and Control, 2003, vol. 1, pp. 556–563

2 Belta, C., and Kumar, V.: 'Abstraction and control for groups of robots', *IEEE Trans. Robo.*, 2004, **20**, pp. 865–875

3 Bowyer, R.S., and Bogner, R.E.: 'Agent behaviour and capability correlation factors as determinants in fusion processing'. Proc. Fusion 2003-Special Session on Fusion by Distributed Cooperative Agents, Cairns, Australia, 2003

4 Das, A., Fierro, R., Kumar, V., and Ostrowski, J.P.: 'A vision-based formation control framework', *IEEE Trans. Robot. Autom.*, 2002, **18**, pp. 813–825

5 Eren, T., Anderson, B.D.O., Morse, A.S., Whiteley, W., and Belhumeur, P.N.: 'Operations on rigid formations of autonoumous agents', *Commun. Inform. Syst.*, 2004, **3**, (4), pp. 223–258

6 Lin, Z., Francis, B.A., and Maggiore, M.: 'Necessary and sufficient graphical conditions for formation control of unicycles', *IEEE Trans. Autom. Control*, 2005, **50**, pp. 121–127

7 Olfati-Saber, R., and Murray, R.M.: 'Graph rigidity and distributed formation stabilization of multi-vehicle systems'. Proc. IEEE Conf. on Decision and Control, 2002, vol. 3

8 Ren, W., and Beard, R.W.: 'A decentralized scheme for spacecraft formation flying via the virtual structure approach', *AIAA J. Guid., Control Dyn.*, 2004, **27**, (1), pp. 73–82

9 Tanner, H.G., Pappas, G.J., and Kumar, V.: 'Leader-to-formation stability', *IEEE Trans Robot. Autom.*, 2004, **20**, (3), pp. 443–455

10 Hendrickx, J.M., Anderson, B.D.O., Blondel, V.D., and Delvenne, J.-C.: 'Directed graphs for the analysis of rigidity and persistence in autonomous agent systems', *Int. J. Robust Nonlinear Control*, in press, 2007

11 Yu, C., Hendrickx, J.M., Fidan, B., Anderson, B.D.O., and Blondel, V.D.: 'Three and higher dimensional autonomous formations: rigidity, persistence, and structural persistence', *Automatica*, 2007, **43**, (3), pp. 387–402

12 Tay, T., and Whiteley, W.: 'Generating isostatic frameworks', *Struct. Topol.*, (11)1985, pp. 21–69

13 Whiteley, W.: 'Rigidity and scene analysis' in Goodman, J., and O'Rourke, J. (Eds.): 'Handbook of discrete and computational geometry' (CRC Press), pp. 893–916

14 Whiteley, W.: 'Some matroids from discrete applied geometry' in Bonin, J.E., Oxley, J.G., and Servatius, B. (Eds.): 'Matroid theory' (American Mathematical Society, 1996), Vol. 197 of Contemporary Mathematics, pp. 171–311

15 Laman, G.: 'On graphs and rigidity of plane skeletal structures', *J. Eng. Math.*, 1970, **4**, pp. 331–340

16 Sandeep, S., Fidan, B., and Yu, C.: 'Decentralized cohesive motion control of multi-agent formations'. Proc. 14th Mediterranean Conf. on Control and Automation, 2006

17 Foulds, L.R.: 'Graph theory applications' (Springer-Verlag, New York, 1992)

18 Eren, T., Goldenberg, D.K., Whiteley, W., Yang, Y.R., Morse, A.S., Anderson, B.D.O., and Belhumeur, P.N.: 'Rigidity, computation, and randomization in network localization'. Proc. INFOCOM Conf. on IEEE Computer and Communications Societies, 2004, vol. 4, pp. 2673–2684