Persistent Autonomous Formations and Cohesive Motion Control

Barış Fidan, Brian D. O. Anderson, Changbin Yu, and Julien M. Hendrickx

CONTENTS
8.1 Introduction ........................................................................................................... 248
8.2 Rigid and Persistent Formations ........................................................................... 250
  8.2.1 Rigid Formations ............................................................................................... 251
  8.2.2 Constraint-Consistent and Persistent Formations ............................................ 252
8.3 Acquisition and Maintenance of Persistence ....................................................... 255
  8.3.1 Acquiring Persistence ....................................................................................... 255
  8.3.2 Persistent Formation Reconfiguration Operations ............................................ 259
  8.3.3 Maintaining Persistence During Formation Reconfigurations ....................... 260
8.4 Cohesive Motion of Persistent Formations ......................................................... 263
  8.4.1 Problem Definition ........................................................................................... 263
  8.4.2 Acyclically Led and Cyclically Led Formations .............................................. 264
8.5 Decentralized Control of Cohesive Motion ......................................................... 266
  8.5.1 Control Design .................................................................................................. 266
    8.5.1.1 Control Law for Zero-DOF Agents ............................................................ 266
    8.5.1.2 Control Law for One-DOF Agents ............................................................ 267
    8.5.1.3 Control Law for Two-DOF Agents ............................................................ 268
  8.5.2 Stability and Convergence ............................................................................... 269
    8.5.2.1 Acyclically Led Minimally Persistent Formations .................................... 269
    8.5.2.2 Cyclically Led Minimally Persistent Formations ...................................... 270
  8.5.3 More Complex Agent Models ........................................................................... 271
8.6 Discussions and Future Directions ...................................................................... 273
Acknowledgment ........................................................................................................ 273
References .................................................................................................................... 274
8.1 Introduction

Recently, the topic of distributed motion control of autonomous multiagent systems has gained significant attention, in parallel with the interest in the real-life applications of such systems involving teams of unmanned aerial and ground vehicles, combat and surveillance robots, underwater vehicles, and so on. [1–4,6,12,20,22–24,26]. This topic presents numerous aspects to be explored corresponding to different control tasks of interest, control approaches to be followed, assumed agent dynamics and interagent information structures, and so on.

In this chapter, using a recently developed theoretical framework of graph rigidity and persistence, we analyze a general class of autonomous multiagent systems moving in formation, namely persistent formations, where the formation shape is maintained during any continuous motion via a set of constraints on each agent to keep its distances from a prespecified group of other neighboring agents constant. As the title indicates, the chapter focuses on two complementary issues about autonomous formations: persistence (which will be explained further below) in Sections 8.2 and 8.3, and cohesive, that is, shape-preserving motion, in Sections 8.4 and 8.5. Before listing the contents and contributions of the chapter and linking these two topics, we give an intuitive introduction to the fundamental terms to be used throughout the chapter. The formal definitions of these terms (where needed) will be given later in the chapter.

We use the term formation for a collection of agents moving in real two- or three-dimensional space to fulfill certain mission requirements. Leaving the agent dynamics issues to future studies in the field and focusing on the motion of the entire formation rather than individual agent behaviors,1 we assume a point-agent system model [14,31]. We represent each multiagent formation F by a graph $G_F = (V_F, E_F)$ with a vertex set $V_F$ and an edge set $E_F$, where each vertex $i \in V_F$ corresponds to an agent $A_i$ in $F$ and each edge $(i, j) \in E_F$ corresponds to an information link between a pair $(A_i, A_j)$ of agents. $G_F$ is also called the underlying graph of the formation $F$. Here, $G_F$ for a particular $F$ can be directed or undirected depending on the properties of information links of $F$, as will be discussed below.

A formation $F$ with an underlying graph $G_F = (V_F, E_F)$ is called rigid if by explicitly maintaining distances between all the pairs of agents which are connected by an information link, that is, whose representative vertices are connected by an edge in $E_F$, the distances between all other pairs of agents in $F$ are consequentially held fixed as well, and hence $F$ can move as a cohesive whole. Typically the agent pairs in $F$ whose interdistances are explicitly maintained are the ones having information (i.e., sensing and communication.

---

1 It is worth noting here that agent dynamics and dynamic interactions are major issues in real world multivehicle formation control and some further discussions on these issues can be found in Reference [25] and the references therein.
links in between, corresponding to the edges in the underlying graph $G_F$. Hence, in (a geometric representation of) the underlying graph $G_F$, explicit maintenance of the distance between an agent pair $(A_i, A_j)$ with an information link between the two corresponds to keeping the length of the edge $(i, j) \in E_F$ constant.

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 $A_i$ and agent $A_j$ at a desired value $d_{ij}$, there is a joint effort of both agent $A_i$ and agent $A_j$ to simultaneously and actively maintain their relative positions. The associated undirected underlying graph will have an undirected edge $(i, j)$ between vertices $i$ and $j$. If enough agent pairs explicitly maintain distances, all remaining interagent distances will be consequently maintained and the formation will be rigid.

In the asymmetric case, which is the assumed control structure in this chapter, only one of the agents in each pair, for example, agent $A_i$, actively maintains its distance to agent $A_j$ at the desired value $d_{ij}$. This means that only agent $A_i$ has to receive the position information broadcast by agent $A_j$, or sense the position of agent $A_j$ and it can make decisions on its own. Therefore, in the asymmetric case, both the overall control complexity and the communication complexity in terms of message sent or information sensed for the formation are expected to be reduced by half. This is modeled in the associated (directed) underlying graph $G_F = (V_F, E_F)$ by a directed edge $(i, j) \in E_F$ from vertex $i$ to vertex $j$. In this case, we also say that $A_i$ has the constraint of staying at a distance $d_{ij}$ from $A_j$ or $A_j$ follows $A_i$ or $A_i$ is a follower of $A_j$.

For a formation $F$ with asymmetric control structure, if each agent in $F$ is able to satisfy all the constraints on it provided that all other agents within $F$ are trying to satisfy their constraints (i.e., satisfy as many of their constraints as possible), then $F$ is called constraint consistent (examples of both a constraint-consistent formation and a formation lacking constraint consistency will be presented subsequently). A formation that is both rigid and constraint-consistent is called persistent [31]. In a persistent formation, 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. For a given persistent formation $F$, if removal of any single edge (in the underlying graph) makes $F$ nonpersistent then $F$ is further called minimally persistent, that is, a minimally persistent formation provably preserves its persistence with a minimal number of edges.

Persistence appears to be the crucial property of an information/control architecture of a formation that ensures that the formation can move.

---

3There exists an exceptional small class of formations in $\mathbb{R}^3$, for which the intuitive explanation here and the formal definition of persistence given in Section 8.2 do not match. This special class is further discussed in Section 8.2.2.
cohesively. Minimal persistence defines those situations where loss of a link means cohesiveness of the motion is no longer assured; from an operational point of view, nonminimal persistence may be desirable to secure redundancy [8].

In Section 8.2, we review the general characteristics of rigid and persistent formations using a recently established framework of rigid and persistent graphs. We present some operational criteria to check the persistence of a given formation. Based on these characteristics and criteria, in Section 8.3, we focus on the acquisition and maintenance of the persistence of certain types of autonomous formations. We particularly consider systematic construction of provably persistent two-dimensional formations by assigning directions to their information links. We briefly review some common operations on persistent formations, including addition of new agents to the formation, closing ranks when an agent is lost, merging two or more formations, splitting a formation into smaller formations, and we provide strategies for maintaining persistence during these operations.

Finally, in Sections 8.4 and 8.5, we focus on cohesive motion control of persistent autonomous formations. We present a set of distributed control schemes to move a given two-dimensional persistent formation with specified initial position and orientation to arbitrary desired final position and orientation without deforming the shape of the formation during the motion. The control design procedure is presented assuming a velocity integrator agent model that is widely considered in the literature [1,12,26]; nevertheless, generalization of these designs for other kinematic models is discussed briefly as well. The chapter concludes with some mention of relevant future research directions.

8.2 Rigid and Persistent Formations

In this section, we give formal definitions of the rigidity and persistence notions and present a brief review of the fundamental characteristics of rigid and persistent formations to the extent needed for the analysis in the following sections. For details the reader may refer to Reference [6,14,28,29,31]. We focus on formations in $\mathbb{R}^2$ and $\mathbb{R}^3$ (two-dimensional and three-dimensional Euclidean spaces, respectively) considering real-world multivehicle formation applications, although most of the definitions and results can be generalized for arbitrary dimensional space $\mathbb{R}^n \ (n \in \{2, 3, \ldots\})$ [31].

Consider a formation $F$ with asymmetric control structure. The directed underlying graph $G^u_F = (V_F, E^u_F)$ of $F$ has been defined in Section 8.1. The undirected graph $G^u_F = (V_F, E^u_F)$ with the same vertex set $V_F$ and the undirected edge set $E^u_F$ having the same edges in $E^u_F$ but with the directions neglected, that is, satisfying

$$(i, j) \in E^u_F \iff ((i, j) \in E^u_F \text{ or } (j, i) \in E^u_F)$$
is called the **underlying undirected graph** of the formation $F$ (or of $G_F$). Next, we focus on formal definition and characterization of rigidity and persistence of formations with asymmetric control structure, using their undirected and directed underlying graphs.

### 8.2.1 Rigid Formations

We formally call a formation $F$ in $\mathbb{R}^n$ ($n \in \{2, 3\}$) with asymmetric control structure **rigid** (and its directed underlying graph $G_F$ **generically n-rigid**) if its undirected underlying graph $G^*_F$ is **generically n-rigid**, where generic n-rigidity of an undirected graph ($n \in \{2, 3\}$) is defined in the sequel. In $\mathbb{R}^n$ ($n \in \{2, 3\}$), a representation of an undirected graph $G = (V, E)$ with vertex set $V$ and edge set $E$ is a function $p : V \rightarrow \mathbb{R}^n$. We say that $p(i) \in \mathbb{R}^n$ 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 $\bar{d}$ for $G$ is a set of distances $d_{ij} > 0$, defined for all edges $(i, j) \in E$. Given a graph $G = (V, E)$ and a corresponding distance set $\bar{d}$, the pair $(G, \bar{d})$ can be considered as a weighted graph [weighted version of the graph $G = (V, E)$], where the weight of each edge $(i, j) \in E$ is $d_{ij} \in \bar{d}$. 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 $\delta(p, p') < \epsilon$, 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). An undirected graph is said to be **generically n-rigid** or **simply n-rigid** ($n \in \{2, 3\}$) if almost all its representations in $\mathbb{R}^n$ are rigid. Some discussions on the need for using the qualifiers "generic" and "almost all" can be found in References [14,27]. One reason for using these terms is to avoid the problems arising from having three or more collinear vertices in $\mathbb{R}^2$ or four or more coplanar vertices in $\mathbb{R}^3$.

Another notion used in rigidity analysis is **minimal rigidity**. A graph $G$ is called **minimally n-rigid** ($n \in \{2, 3\}$) if $G$ is n-rigid and if there exists no n-rigid subgraph of $G$ with the same set of vertices as $G$ and a smaller number of edges than $G$. Provably equivalently, a graph is **minimally n-rigid** if it is n-rigid and if no single edge can be removed without losing n-rigidity. Fundamental characteristics of rigid and minimally rigid graphs and some of their applications

---

In this chapter, we use $| \cdot |$ to denote two different operators, one for vectors and the other for sets. For a vector $\xi \in \mathbb{R}^n$ ($n \in \{2, 3\}$), $|\xi|$ denotes the Euclidean norm of $\xi$. Hence $|p_1(i) - p_2(i)|$, on this page, denotes the Euclidean distance between $p_1(i)$ and $p_2(i)$. For a set $S$, $|S|$ denotes the number of elements of $S$. 

---

---
in autonomous formation control can be found in References [6,27–29]. Following are a selection of these characteristics.

**THEOREM 1**
For any n-rigid graph $G = (V, E)$ ($n \in \{2, 3\}$) with at least $n$ vertices, there exists a subset $E' \subseteq E$ of edges such that the graph $G' = (V, E')$ is minimally n-rigid and satisfies the following: (1) $|E'| = n|V| - n(n + 1)/2$; (2) any subgraph $G'' = (V'', E'')$ of $G'$ with at least $n$ vertices satisfies $|E''| \leq n|V''| - n(n + 1)/2$.

**LEMMA 1**
Let $G = (V, E)$ be a minimally n-rigid graph ($n \in \{2, 3\}$) and $G' = (V', E')$ be a subgraph of $G$. If $|E'| = n|V'| - n(n + 1)/2$ then $G'$ is minimally n-rigid.

**LEMMA 2**
For $n \in \{2, 3\}$, a graph obtained by adding one vertex to a graph $G = (V, E)$ and $n$ edges connecting this vertex to other vertices of $G$ is (minimally) n-rigid if and only if $G$ is (minimally) n-rigid.

### 8.2.2 Constraint-Consistent and Persistent Formations

Similar to the definition of rigid formations, we call a formation $F$ in $\mathbb{R}^n$ ($n \in \{2, 3\}$) with asymmetric control structure persistent (constraint consistent) if its directed underlying graph $G_F$ is n-persistent (respectively n-constraint consistent), where n-persistence and n-constraint consistency of a directed graph are defined as follows.

Consider a directed graph $G = (V, E)$, a representation $p : V \rightarrow \mathbb{R}^n$ ($n \in \{2, 3\}$) of $G$, and a set of desired distances $d_{ij} > 0$, $\forall (i, j) \in E$. Note here that the representation, vertex positions, and distance between two representations corresponding to a directed graph are defined exactly the same as the ones corresponding to undirected graphs (see Section 8.2.1). 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 $p^* \in \mathbb{R}^n$ for which

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

A representation $p$ of a graph is called fitting 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. The representation $p$ is called 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
Persistent Autonomous Formations and Cohesive Motion Control

Application of Theorem 2 in $\mathbb{R}^2$. Assume that the distance set $d$ is given by $d_{12} = d_{13} = d_{23} = d_{25} = d_{31} = d_{45} = 1$ and for [b] and [c] $d_{24} = \sqrt{2}$. (a) The representation is constraint consistent but not rigid. (Assuming vertices [agents] 1, 2, and 3 are stationary, 4 and 5 can continuously move to new locations without violating $d_{25}$, $d_{34}$, $d_{45}$.) Hence it is not persistent. (b) The representation is rigid but not constraint consistent. (Again assuming that 1, 2, and 3 are stationary, 5 can continuously move to new positions without violating the distance constraint $[d_{25}]$ on it, for which vertex 4 is unable to meet all three distance constraints $[d_{24}, d_{34}, d_{45}]$ on it at the same time.) Hence it is not persistent. (c) The representation is both rigid and constraint consistent, hence it is persistent.

generically $n$-persistent ($n \in \{2, 3\}$) if almost all its representations in $\mathbb{R}^n$ are persistent.

Similarly, a representation $p$ is called constraint consistent if there exists $\epsilon > 0$ such that any representation $p'$ fitting for the distance set $\delta$ induced by $p$ and satisfying $\delta(p, p') < \epsilon$ is a realization of $\delta$. Again, we say that a graph is generically $n$-constraint consistent ($n \in \{2, 3\}$) if almost all its representations in $\mathbb{R}^n$ are constraint consistent. The relation among persistence, rigidity, and constraint consistency of a directed graph is given in the following theorem and demonstrated using a two-dimensional example in Figure 8.1.

THEOREM 2
[31] A representation in $\mathbb{R}^n (n \in \{2, 3\})$ is persistent if and only if it is rigid and constraint consistent. A graph is generically $n$-persistent ($n \in \{2, 3\}$) if and only if it is generically $n$-rigid and generically $n$-constraint consistent.

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 $G$, that is, the number of edges in $G$ heading to and originating from $i$, respectively.

PROPOSITION 1
[31] An $n$-persistent graph ($n \in \{2, 3\}$) remains $n$-persistent after deletion of any edge $(i, j)$ for which $d^{+}(i) \geq n + 1$. Similarly, an $n$-constraint-consistent graph ($n \in \{2, 3\}$) remains $n$-constraint consistent after deletion of any edge $(i, j)$ for which $d^{+}(i) \geq n + 1$.

---

1Rigidity for a directed graph is defined in the same way as for an undirected graph; one simply takes no account of any assigned direction.
Another notion found useful in characterizing a persistent formation $F$ (or its underlying graph $G_F$) is the number of degrees of freedom (DOF count) of each agent (vertex) in $\mathbb{R}^n$ ($n \in \{2, 3\}$), which is defined as the maximal dimension, over all $n$-dimensional representations of $G_F$, of the set of possible fitting positions for this agent (vertex). In $\mathbb{R}^n$ ($n \in \{2, 3\}$), the vertices with zero out-degrees have $n$ DOFs, the vertices with out-degree 1 have $n - 1$ DOFs, the ones with out-degree 2 have $n - 2$ DOFs, and all the other vertices have zero DOF. In an underlying graph of an $n$-dimensional formation ($n \in \{2, 3\}$), a vertex (agent) with $n$-DOF in $\mathbb{R}^n$ is also called a leader. The following corollary of Proposition 1 provides a natural bound on the total number of degrees of freedom in an $n$-persistent graph in $\mathbb{R}^n$ ($n \in \{2, 3\}$), which we also call the total DOF count of that graph in $\mathbb{R}^n$.

**COROLLARY 1**
The total DOF count of an $n$-persistent graph in $\mathbb{R}^n$ ($n \in \{2, 3\}$) can at most be $n(n + 1)/2$.

In Corollary 1, note that $n$ of the $n(n + 1)/2$ DOFs correspond to translations and the remaining $n(n - 1)/2$ correspond to rotations of a formation represented by the $n$-persistent graph, considering the whole formation as a single body. Next, we present two other essential results on the characterization of persistent graphs (and hence persistent formations), proofs of which can be found in Reference [31].

**THEOREM 3**
A directed graph is $n$-persistent ($n \in \{2, 3\}$) if and only if all those subgraphs are $n$-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 $n$.

**PROPOSITION 2**
[31] Consider a directed graph $G$ and another directed graph $G'$ that are obtained by adding one vertex to $G$ and at least $n$ edges leaving this vertex and incident on different vertices of $G$ ($n \in \{2, 3\}$). Then, $G'$ is $n$-persistent if and only if $G$ is $n$-persistent.

It has been stated in Section 8.1 that there exists a particular small class of formations in $\mathbb{R}^3$, for which the intuitive definition of persistence given in Section 8.1 and the formal definition, here in Section 8.2, do not match. The problem in this exceptional class arises when it is not possible for all the agents in a certain subset of the agent set of the formation to simultaneously satisfy all their constraints, despite the ability of any single agent to move to a position that satisfies the constraints on it once all the other agents are fixed [31]. Persistent formations free of this problem are called structurally persistent. For a formal definition and characteristics of structural persistence, as well as the details of the above problem, the reader may refer to Reference [31]. The distinction between structural persistence and persistence does not arise in two dimensions. In $\mathbb{R}^3$, it turns out that a formation is structurally persistent.
if and only if it is persistent and does not have two leaders each with three DOFs. For simplification, we assume all the practical persistent formations considered in the chapter to be structurally persistent as well.

8.3 Acquisition and Maintenance of Persistence

8.3.1 Acquiring Persistence

The importance of persistence for cohesive and coordinated motion of autonomous formations with asymmetric control structure has been indicated in the previous sections. In this subsection, we focus on acquisition of persistence for such formations, which can be interpreted in various ways: (1) systematic construction of a persistent formation from a given team of autonomous agents with certain desired information architecture characteristics; (2) converting a given nonpersistent or non-rigid formation to a persistent one via swapping some of the directions of the information links and adding some extra links if needed; (3) assigning directions to the links of formations with given rigid undirected underlying graphs (i.e., to the edges of the undirected underlying graphs) to obtain persistent formations; and so on.

Interpretation (1) is partially analyzed in References [14–16,31], where certain systematic construction procedures have been developed similar to their well-established counterparts for growing undirected rigid formation (or graphs), namely Henneberg construction sequences [6,27]. In References [15,16], a systematic procedure is developed for constructing (minimally) two-persistent graphs, where at each step of the procedure, one of the following three operations is applied: vertex addition, edge splitting, and edge reversal. Each of these operations (as defined below) preserves minimal two-persistence when applied to a minimally two-persistent graph and two-persistence when applied to a two-persistent graph. Hence, if the procedure starts with a two-persistent graph \( G_0 \), the graph \( G_i \) obtained at each step \( i = 1, 2, \ldots \) is two-persistent; and if \( G_0 \) is further a minimally two-persistent graph, each \( G_i \) is minimally two-persistent. Next, we briefly explain the three operations.

At step \( i (i \in \{1, 2, \ldots \}) \), application of a vertex addition to graph \( G_{i-1} = (V_{i-1}, E_{i-1}) \) means addition to \( G_{i-1} \) of a vertex \( j \) with in-degree 0 and out-degree 2 and two distinct edges

\[
(j, k), (j, l)
\]

outgoing from \( j \) where \( k, l \in V_{i-1} \). The resultant graph is \( G_i = (V_i, E_i) \) where

\[
V_i = V_{i-1} \cup \{j\}
\]

and

\[
E_i = E_{i-1} \cup \{(j, k), (j, l)\}
\]
Application of edge splitting means removing a directed edge
\[(j, k) \in E_{i-1}\]
and adding a new vertex \(l\) and the edges
\[(j, l), (l, k), (l, m)\]
where \(m \in V_{i-1}\), that is, the resultant graph is \(G_i = (V_i, E_i)\) where
\[V_i = V_{i-1} \cup \{l\}\]
and
\[E_i = E_{i-1} \cup \{(j, l), (l, k), (l, m)\} \setminus \{(j, k)\}\]
Finally, application of edge reversal on \(G_{i-1}\) is replacing directed edge
\[(j, k) \in E_{i-1}\]
where \(j \in V_{i-1}\) has at least 1 DOF (in \(\mathbb{R}^2\)) with \((k, j)\) to obtain \(G_i = (V_i, E_i)\) with \(V_i = V_{i-1}\) and
\[E_i = E_{i-1} \cup \{(k, j)\} \setminus \{(j, k)\}\]
Any minimally two-persistent graph \(G = (V, E)\) with \(V = \{1, 2, \ldots, N\}\), where \(|V| = N \geq 3\), can be obtained starting with a seed graph \(G_0 = (V_0, E_0)\) with
\[V_0 = \{1, 2, 3\}, E_0 = \{(2, 1), (3, 1), (3, 2)\}\]
and applying the procedure described above in the following particular form [15,16]: First, using a sequence of \(N - 3\) operations each of which is either vertex addition or edge splitting, a minimally two-persistent graph \(G_{N-3}\) is built having the same underlying undirected graph as \(G\). Then from \(G_{N-3}\), a graph \(G' = (V, E')\) is obtained having the same undirected underlying graph and the same DOF distribution (among vertices in \(V\) in \(\mathbb{R}^2\)) with \(G\), by redistributing the DOFs among the vertices by applying a sequence of edge reversals. It is further shown in References [15,16] that the only possible differences between \(G'\) and \(G\) are directions (orientations) of certain cycles. \(G\) is obtained by reversing the directions of these cycles via a sequence of edge reversals for each cycle, that is, reversing the direction of each of the edges in these cycles in a sequential manner, as the final part of the construction procedure. The doability of each of the three parts of the above procedure for building \(G\) from \(G_0\) is proven in References [15,16]. As a special case, if \(G\) is an acyclic (cycle free), minimally two-persistent graph, then \(G\) (with possibly a different permutation of vertex indices) can be grown from \(G_0\) by applying a sequence of \(N - 3\) vertex additions [14].
A generalized version of the vertex addition operation above (for both two-persistence and three-persistence) is discussed in References [14,31], where a vertex with out-degree at least \( n \) and in-degree 0 is added to an \( n \)-persistent graph \((n \in \{2, 3\})\). The results in References [14,31] imply that the generalized vertex addition operation preserves \( n \)-persistence \((n \in \{2, 3\})\) and any acyclic (cycle free) \( n \)-persistent graph \( G = (V, E) \) with \( V = \{1, 2, \ldots, N\} \), where \( |V| = N \geq n + 1 \) can be grown from an acyclic \( n \)-persistent seed graph with three vertices, performing a sequence of \( \bar{N} - 3 \) generalized vertex additions.

One possible approach to the interpretation (2) of persistence acquisition is to perform the acquisition task in two steps, where in the first step the undirected underlying graph is made rigid via addition of a necessary number of links with arbitrary directions, and in the second step directions of selected links are swapped to satisfy constraint consistency and hence persistence of the directed underlying graph. This interpretation has not been fully analyzed in the literature yet, but partial discussions on or relevant to making a non-rigid graph rigid via adding edges and making a nonconstraint-consistent directed graph constraint consistent via edge reversals can be found in References [6,15,16,29]. Particularly, a discussion on making a nonrigid graph rigid via adding edges is presented in Reference [6], where the task is named as a (minimal) cover problem.

A general solution to the problem defined in interpretation (3), that is, one applicable to nonminimally rigid as well as minimally rigid graphs, is not available in the literature yet, which is not unreasonable given 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. Nevertheless, systematic solutions for classes of nonminimally rigid undirected graphs, namely complete graphs, bilateral and trilateralation graphs, wheel graphs, \( C^2 \) graphs, \( C^3 \) graphs, and bipartite graphs of type \( K_{m,n} \), are provided in Reference [8]. Formal definitions of these graph classes can be found in References [7,10] and their rigidity can be verified easily using these definitions and the rigidity criteria available in the literature, for example, References [27–29]. Below we present the persistence acquisition procedures for some of these classes (depicted in Figure 8.2) followed by some discussion on their practical implications. The complete list of results as well as the proofs can be seen in Reference [8].

**PROPOSITION 3**

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

**PROPOSITION 4**

Given a trilateration graph \( T \), that is, 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
vertices $1, 2, \ldots, j - 1$ for $j = 4, 5, \ldots, k$, let $\overline{T}$ be the directed graph obtained by assigning directions to the edges of $T$ such that the direction of each edge $(i, j)$ for $i < j$ is from $j$ to $i$. Then, $\overline{T}$ is $n$-persistent for $n \in \{2, 3\}$.

**Proposition 5**

Given an integer $k \geq 3$, consider the wheel graph $W_k$ that is composed of $k$ rim vertices, labeled 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 (labeled vertex 0), and the edges $(0, i)$ for $i = 1, 2, \ldots, k$ connecting the hub vertex to each of the rim vertices. Let $\overline{W}_k$ 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, $\overline{W}_k$ is two-persistent.

Note that each of the rigid graph classes considered above corresponds to a formation architecture that can be used in guidance and control of autonomous multiagent formations. Complete graphs model the information architecture of formations where the sensing (communication) radius of each agent potentially allows it to maintain its distance actively from any other agent in the entire formation. Trilateration results given in Proposition 4 can be used in acquisition of cycle-free formations with leader-follower structure.
[26] and asymmetric control architecture. One might use wheel graphs to model two-dimensional formations in which there is a central agent, the commander, which can be sensed by all other agents. Note here that since the sensing/communication capabilities of the agents and the distance constraints on them may be different from each other, the formations with wheel underlying graphs may have various geometries such as the ones depicted in Figure 8.2c and d. As demonstrated in Figure 8.2d, the commander (corresponding to the hub vertex) does not need to be in the geometric center of the formation.

Although the procedures in Reference [8], including the results above, have been developed for a limited number of formation classes, the methodology used to develop these procedures can be used to generate similar procedures for persistence acquisition and of other formation classes as well.

### 8.3.2 Persistent Formation Reconfiguration Operations

In many autonomous multiagent formation applications, some of which are mentioned in Section 8.1, one needs to analyze certain scenarios that have a significant likelihood in practice, as a matter of guaranteeing robustness in the presence of such scenarios. In Reference [6] three key categories of operations on rigid formations have been analyzed: merging, splitting, and closing ranks. The focus of analysis in Reference [6] for each operation is preservation of rigidity during the operation.

The trivial extensions of the above operations for persistent formations and the relevant persistence maintenance problems during these operations can be defined as follows. In **merging** the task is to establish a set $L_n$ of new directed [information] links between (agents of) two persistent formations $F_1$, $F_2$ such that the merged formation $F_1 \cup F_2 \cup L_n$ (i.e., the formation whose agent set is the union of the agent sets of $F_1$ and $F_2$ and whose directed [information] link set is $L_1 \cup L_2 \cup L_n$, where $L_i$ denotes the directed [information] link set of $F_i$ for $i \in \{1, 2\}$) is persistent. In terms of the underlying graphs, the merging task is equivalent to the following: Given the underlying graphs $G_{F_1} = (V_{F_1}, E_{F_1})$, $G_{F_2} = (V_{F_2}, E_{F_2})$ of two persistent formations $F_1$ and $F_2$, find a directed edge set $E_n$ such that the directed graph $(V_{F_1} \cup V_{F_2}, E_{F_1} \cup E_{F_2} \cup E_n)$ is persistent.

In **splitting**, which can be thought of as the reverse of merging, the case is considered where a persistent formation $F$ with directed underlying graph $G_F = (V_F, E_F)$ is split into two formations $F_1$, $F_2$ with directed underlying graphs $G_{F_1} = (V_{F_1}, E_{F_1})$, $G_{F_2} = (V_{F_2}, E_{F_2})$, respectively (where $V_F = V_{F_1} \cup V_{F_2}$) due to loss of some information links in $F$ (or some edges in $E_F$). The task is to establish new links within each of $F_1$, $F_2$ (add new directed edges to $E_{F_1}, E_{F_2}$) such that both $F_1$ and $F_2$ become persistent.

**Closing ranks** can be thought of as a special (pseudo-) splitting operation. The case of interest is the loss of an agent (and the links associated to this agent) from a persistent formation, and the closing ranks task is to establish new directed links between certain pairs among the remaining agents such that the
new formation (formed after the agent loss and establishment of the new links) is persistent as well. Note here that splitting can be thought of as a generalized closing ranks operation (defined for the loss of a set of agents instead of a single agent) as well, observing that the scenario of the above splitting problem for the post-split formation $F_1$, for example, can be equivalently reformulated as $F_1$ being what is left when $F$, having initially $F_2$ as its subformation, then loses the agents in the subformation $F_2$. This observation has been found useful (at least for the undirected underlying graphs and rigidity considerations) in treating splitting problems using certain results derived for the closing ranks problem, for example, in Reference [6].

The persistence maintenance problems corresponding to the three operations above can be thought of as special cases of the problem of making a nonpersistent (underlying) graph persistent via adding some new edges (and swapping some of the edge directions), which can be thought of as the extension of the (minimal) cover problem discussed in Reference [6] for directed graphs or formations. A complete generalization to directed graphs of the solution to the minimal cover problem has not yet been achieved. Moreover, the three operations and the corresponding persistence maintenance tasks can be further generalized to consider merging involving more than two persistent formations, splitting into more than two persistent formations, closing ranks during loss of two or more agents, and so on. Furthermore, various other scenarios can be generated as combinations of specific forms of a number of the above formations. One such scenario is where a multiagent (vehicle) formation loses some of its agents and new agents are required to be added to the formation without violating the existing control structure [6]. Another similar scenario is where the leader of a formation has to be substituted due to evolving mission requirements [8,30]. If the formation in the beginning is persistent, the leader change task above without damaging the control structure can be abstracted as changing directions of certain edges in an underlying directed graph in an appropriate way that maintains the persistence.

We conclude this subsection with a real-life example where frequent formation changes are expected: terrain surveillance using a formation of aerial vehicles with surveying sensors mounted on them [3,8]. In this application, 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. In such a case it is essential to coordinate well the behavior of each such additional sensor agent with that of the already existing agents, which can be done by maintaining persistence of the formation during variations.

8.3.3 Maintaining Persistence During Formation Reconfigurations

As mentioned in Section 8.3.2, the main issue in the reconfiguration operations on persistent formations in terms of the information architecture is maintenance of persistence. Full analysis of the basic or extended versions of the merging, splitting, and closing ranks operations for persistent formations.
and the relevant persistence maintenance problems are not available in the literature yet. However, some partial analysis results and relevant discussions can be found in some recent studies, for example, in References [8,17,30,31].

In Reference [17], the problem of merging persistent formations in \( \mathcal{R}^2 \) and \( \mathcal{R}^3 \) is partially analyzed in a so-called metaformation framework. The main results of this analysis are summarized in the following theorems. The first theorem is about a particular way of merging the directed underlying graphs of persistent formations via addition of a set \( E_M \) of additional edges having end-vertices belonging to different underlying graphs; and the second is about necessary and sufficient conditions for an edge-optimal persistent merging, that is, for having \( E_M \) such that no single edge of \( E_M \) can be removed without losing persistence of the merged formation.

**THEOREM 4**

[17] A collection of \( n \)-persistent graphs \( G_1, \ldots, G_N \) \((n \in \{2, 3\}, \ N \in \{2, 3, \ldots\}) \) can be merged into a (structurally) \( n \)-persistent graph if and only if this merging can be done by adding edges leaving vertices with one or more local DOFs in \( \mathcal{R}^n \), that is, DOFs (in \( \mathcal{R}^n \)) in the corresponding \( G_i \) \((i \in \{1, 2, \ldots, N\}) \), such that the original local DOF count of each vertex is greater than or equal to the number of added edges leaving this vertex. In that case, the merged graph \( G \) is \( n \)-persistent if and only if it is \( n \)-rigid. \( G \) is then structurally three-persistent if and only if it has at most one vertex with three DOFs in \( \mathcal{R}^3 \).

**THEOREM 5**

[17] Consider a set \( G = G_3 \cup G_2 \cup G_1 \) of disjoint directed graphs where \( G_3 \) is composed of \( n \)-persistent graphs \((n \in \{2, 3\}) \) having at least three vertices, \( G_2 \) is composed of directed graphs with two vertices and an edge, and \( G_1 \) is composed of single-vertex graphs. Then, \( G = \bigcup_{G_i \in G} G_i \cup E_M \), where \( E_M \) is a set of additional edges having end-vertices belonging to different graphs in \( G \), is an edge-optimal persistent merging in \( \mathcal{R}^n \) if and only if the following conditions all hold:

1. \[ |E_M| = n(n + 1)/2(|G_3| - 1) + (2n - 1)|G_2| + n|G_1|. \]
2. For all nonempty \( E'_M \subseteq E_M \), there holds \[ |E'_M| \leq n(n + 1)/2(|I(E'_M)| - 1) + (2n - 1)|I(E'_M)| + n|K(E'_M)| \], where \( I(E'_M) \) is the set of graphs in which at least three vertices or two unconnected ones are incident to edges of \( E'_M \), \( I(E'_M) \) is the set of those in which one connected pair of vertices is incident to edges of \( E'_M \), and \( K(E'_M) \) is the set of those in which only one vertex is incident to edge(s) of \( E'_M \).
3. All edges of \( E_M \) leave vertices with local DOFs in \( \mathcal{R}^n \).

In References [8,31], persistence maintenance of a three-dimensional formation is analyzed during addition of a new agent to the formation using a DOF allocation state framework. This framework is based on the elaboration of the fact that in a three-persistent graph, there are at most six DOFs (in \( \mathcal{R}^3 \)) to be allocated among the vertices [8,31]. The six DOF allocation states in the framework are defined as the following six sets of DOF counts (in \( \mathcal{R}^3 \)) of
vertices ordered in a nonincreasing manner, which represent the six different ways (considering the agents as indistinguishable) of allocating the six DOFs, where in the state $S_1$, for example, one vertex (the leader) has three DOFs, one has two DOFs, another has one DOF, and all the others are 0-DOF: 

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

Further discussion on the DOF allocation states can be found in Reference [8]. Employing an analysis based on the DOF allocation framework, a set of directed vertex addition operations requiring a minimal number of new edges, namely directed trilateration operations, have been developed in Reference [31] for maintaining persistence. A directed trilateration, $\text{DT}(m)$, where $m \in \{0, 1, 2, 3\}$, is defined as a transformation of a three-persistent graph $G = (V, E)$, where $|V| \geq 3$, to another three-persistent graph $G' = (V', E')$, where

\[ V' = V \cup \{i\}, \quad E' = E \cup \{(i, k) : \forall k \in V_1\} \cup \{(j, l) : \forall j \in V_2\} \]

for some $V_1, V_2 \subseteq V$ satisfying $V_1 \cap V_2 = \emptyset$, $|V_1| = 3 - m$, $|V_2| = m$, and $\text{DOF}(j) \geq 1, \forall j \in V_2$, provided that the vertices of $V_1 \cup V_2$ are all distinct and are not collinear [31].

Note here that Theorem 3 indicates that the graph obtained after applying a directed trilateration is three-persistent, that is, the directed trilateration defined above preserves the three-persistence of the graphs. Furthermore, an undirected graph formed by applying a sequence of trilateration operations starting with an initial undirected triangle, often called a trilateration graph, is guaranteed to be generically three-rigid [28,29]. Similarly, 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 three-persistent [8,31]. Further properties and interpretations of the four directed trilateration operations $\text{DT}(0)$, $\text{DT}(1)$, $\text{DT}(2)$, and $\text{DT}(3)$ are given in Reference [8].

**REMARK 1**

In the implementation of the persistence acquisition and maintenance strategies presented in this chapter, a common requirement would be developing decentralized controllers for individual agents, instead of a centralized 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.

---

5 If there is no such $V_2$ then the corresponding $\text{DT}(m)$ cannot be performed for the graph $G$.}
8.4 Cohesive Motion of Persistent Formations

8.4.1 Problem Definition

In the previous sections we have focused on characteristics of persistent autonomous formations and discrete procedures to acquire and maintain persistence, without considering any dynamic control task required for the formation. In this section and the next, we focus on decentralized motion control of two-dimensional persistent formations where each of the agents makes decisions based only on its own observations and state. The particular problem we deal with in these two sections, in its basic form, is how to move a given persistent formation with specified initial position and orientation to a new desired position and orientation cohesively, that is, without violating the persistence of the formation during the motion. More specific definition of the problem is given as follows.

**PROBLEM 1**

Consider a persistent two-dimensional formation $F$ with $m \geq 3$ agents $A_1, \ldots, A_m$ whose initial position and orientation in $\mathbb{R}^2$ (the $xy$-plane) are specified with a set $d$ of desired inter-agent distances $d_{ij}$ between neighbor agent pairs $(A_i, A_j)$, that is, the initial position $p_{i0}$ of each agent $A_i$ ($i \in \{1, \ldots, m\}$) is known, where the initial positions $p_{i0}$ are consistent with $d$. The control task is to move $F$ to a given desired (final) position and orientation defined by a set of final positions $p_{if}$ of the individual agents $A_i$ for $i = 1, \ldots, m$, where $p_{if}$ are consistent with $d$, cohesively, that is, without deforming the shape or violating the distance constraints of $F$ during motion, using a decentralized strategy.

We perform our control design and analysis in continuous-time domain with the following assumptions, relaxation or generalization of which will be discussed later:

A1: Each agent $A_i$ has a velocity integrator kinematics

$$p_i(t) = v_i(t)$$  \hspace{1cm} (8.1)

where $p_i(t) = [x_i(t), y_i(t)], v_i(t) = [v_{xi}(t), v_{yi}(t)] \in \mathbb{R}^2$ denote the position and velocity of $A_i$ at time $t$, respectively.

A2: The individual controller of each agent $A_i$ ($i \in \{1, \ldots, m\}$) can adjust the velocity $v_i(t)$ directly, that is, $v_i(t)$ is the control signal of agent $A_i$. The controller of $A_i$ ($i \in \{1, \ldots, m\}$) is assumed to guarantee that $v_i(t)$ is continuous and $|v_i(t)| \leq \vartheta, \forall t \geq 0$ for some constant maximum speed limit $\vartheta > 0$.

A3: Each agent $A_i$ knows its final desired position $p_{if}$ and can sense its own position $p_i(t)$ and velocity $v_i(t)$ as well as the position $p_j(t)$ of any agent $A_j$ it follows at any time $t \geq 0$. 


A4: The distance-sensing range for a neighbor agent pair \((A_i, A_j)\) is sufficiently larger than the desired distance \(d_{ij}\) to be maintained.

Note here that Problem 1, together with the assumptions A1 to A4, is formulated in a simple form in order to simplify the initial analysis of cohesive motion of persistent formation to be presented in this chapter. In a more realistic or more practical scenario one would need to consider more complex agent dynamics; noise, disturbance, and time delay effects in sensing, control, and communication; imperfect position and distance sensors providing measurements with uncertainties; obstacles in the area of interest that the formation has to avoid; and so on. Moreover, there would be some optimality criteria for the control task in terms of the overall process duration, physical and computational energy consumption, and so on. Again these issues are neglected for the convenience of building a clear initial design and analysis framework that can be elaborated later for a particular, more practical problem according to particular, specifications.

On the other hand, in order to make our design and analysis framework usable in more complex practical scenarios, we need to consider the requirements in possible extensions of Problem 1 and pay attention to simplicity and robustness, even if not needed for the sake of solving Problem 1 only. For example, a straightforward attempt to solve Problem 1 based on predetermining a suitable time trajectory for the formation starting at the given initial setting and ending at the final desired setting, and hence a time trajectory \(p_i(t)\) for each agent \(A_i\), and then generating \(v_i\) for each \(A_i\) that would result in the predetermined trajectory \(p_i(t)\), would not be easily extendible for more complex scenarios; in fact, it might not even be possible.

In our approach, we choose the control laws below so that meeting the distance constraint has a higher priority than reaching to the final desired position which can be rewritten as a guideline as follows:

G1: A 0-DOF agent has to use all of its control potential for satisfying its distance constraints, and a 1-DOF agent can undergo its DOF movement, only when its distance constraint is satisfied within a certain error bound.

8.4.2 Acyclically Led and Cyclically Led Formations

For a complete analysis of Problem 1, we need to take into account the following categorization of two-dimensional persistent formations in terms of distribution of the DOFs. The sum of the DOFs of individual agents in a persistent two-dimensional formation is at most three, and for a minimally persistent formation, exactly three, which is the same as the DOF of a free rigid (nonvertex) object in \(\mathbb{R}^2\) (two for translation and one for rotation) \([14,31]\). Based on the distribution of these three DOFs, minimally persistent formations can be divided into two categories: acyclically led minimally persistent formations (or formations with the leader-follower structure) where one agent
Persistent Autonomous Formations and Cohesive Motion Control

265

has two DOFs, another has one DOF, and the rest have zero DOFs, and cyclically led minimally persistent formations (or formations with the three-coleader structure) where three agents have one DOF and the rest have zero DOF. It can be easily shown that any two-dimensional minimally persistent formation has to be either acyclically led or cyclically led.

In an acyclically led formation, the two-DOF agent is called the leader and the one-DOF agent is called the first follower. In a cyclically led formation, the one-DOF agents are called the coleaders. In both structures the zero-DOF agents are called the (ordinary) followers. The names cyclically led and acyclically led come from the following facts, which can be easily shown using the definition of DOF and Lemma 1 of Reference [15]: There is no cycle in an acyclically led formation passing through the leader; and there always exists a cycle passing through all of the three coleaders in a cyclically led formation.

In a cyclically led formation, because of lying on a cycle, the motions of the three coleaders are cyclically dependent on each other and hence the motion control for the formation requires a more implicit strategy than one for an acyclically led formation. Some stability properties of a subclass of acyclically led formations are investigated in Reference [26]; however, such an investigation is not available yet for cyclically led formations.

In Section 8.5, we design controllers to solve Problem 1 for both the cyclically led and acyclically led categories of minimally persistent formations. Note here that there exist acyclically led (minimally) persistent formations where the first follower does not directly follow the leader but another (ordinary follower) agent (see Figure 8.3a), and there exist cyclically led (minimally) persistent formations where the coleaders do not directly follow each other but some other agents (see Figure 8.3b). For simplicity, in Section 8.5, we assume that the first follower directly follows the leader in an acyclically led formation and the three coleaders follow each other in a cyclically led formation.

![Figure 8.3](image_url)

Typical minimally persistent formations in R^2: (a) An acyclically led formation where the first follower (A_4) does not follow the leader (A_1). (b) A cyclically led formation where the coleaders (A_1, A_3, A_5) do not follow each other.
8.5 Decentralized Control of Cohesive Motion

8.5.1 Control Design

Based on the definition of Problem 1, Assumptions A1 to A4, and Guideline G1, it can be judged that the structure of the individual controller of each agent in the minimally persistent formation of interest should be specific to its DOF. In other words, three types of controllers are needed, one for each zero-DOF, one-DOF, and two-DOF agent sets, regardless of whether the formation is acyclically led or cyclically led, although the motion behaviors and stability and convergence analysis for these two categories are expected to be different.

In our control design to solve Problem 1 for minimally persistent formations, we use basic vector analysis and borrow ideas from the virtual vector field concept, details of which can be found in Reference [5] and the references therein. The main idea in the virtual vector field approach is obtaining the overall velocity vector (for each agent) as the superposition of the vectors defining each of the separate motion tasks (of this agent). In our case the two separate motion vector types of an agent are (1) to maintain a distance constraint with each of the agents it follows and (2) to move towards a final destination.

For optimality considerations and to cope with constant velocity requirements in certain unmanned air vehicle (UAV) and other flight formation applications, we assert the following two additional guidelines in our control design in addition to Assumptions A1 to A4 and Guideline G1:

G2: Any agent shall move at the constant maximum speed \( v > 0 \) at any time instant \( t \geq 0 \) unless it is impossible to do so at that particular instant \( t \) due to, for example, initial and final zero-velocity constraints, and so on.

G3: Any zero-DOF or one-DOF agent shall move through a path of shortest length in order to satisfy its distance constraints.

Based on Assumptions A1 to A4 and Guidelines G1 to G3, we design a control scheme for zero-DOF, one-DOF, and two-DOF agents separately. Note here that the guidelines are labeled as guidelines (separate from the assumptions), because it is recognized that there may be clashes between them. The way these clashes are resolved is embedded in the control laws presented below.

8.5.1.1 Control Law for Zero-DOF Agents

Consider a zero-DOF agent \( A_i \) and the two agents \( A_j \) and \( A_k \) it follows. Note here that, in a two-dimensional minimally persistent formation, any zero-DOF

---

6 However, the virtual vector field approaches described in these works are different from our approach, as in these works, the interagent distance constraints are not considered and hence do not constitute a vector field.
agent necessarily follows exactly two other agents by definition of minimal persistence and Proposition 1. Due to the distance constraints of keeping \(|p_i(t) - p_j(t)|, |p_i(t) - p_k(t)|\) at the desired values of \(d_{ij}, d_{ik}\), respectively, at each time \(t \geq 0\), the desired position \(p_{id}(t)\) of \(A_i\) is the point whose distances to \(p_j(t)\) and \(p_k(t)\) are \(d_{ij}, d_{ik}\), respectively, and which satisfies continuity of \(p_{id}(t)\). Assuming \(|p_i(t) - p_{id}(t)|\) is sufficiently small, \(p_{id}(t)\) can be explicitly determined as:

\[
p_{id}(t) = p_{jk}(t, p_i(t))
\]

(8.2)

where \(p_{jk}(t, p_0)\) for any \(p_0 \in \mathbb{R}^2\) denotes the intersection of the circles \(C(p_j(t), d_{ij})\) and \(C(p_k(t), d_{ik})\) that is closer to \(p_0\), and in the notion \(C(\cdot, \cdot)\) the first argument denotes the center and the second denotes the radius. Based on this observation, we propose the following control law for the zero-DOF agent \(A_i\):

\[
v_i(t) = \delta \beta_i(t) \delta_{id}(t) / |\delta_{id}(t)|
\]

\[
\delta_{id}(t) = p_{id}(t) - p_i(t) = p_{jk}(t, p_i(t)) - p_i(t)
\]

(8.3)

\[
\beta_i(t) = \begin{cases} 
0, & |\delta_{id}(t)| < \varepsilon_k \\
\frac{|v_i(t) - \varepsilon_k|}{\varepsilon_k}, & \varepsilon_k \leq |\delta_{id}(t)| < 2\varepsilon_k \\
1, & |\delta_{id}(t)| \geq 2\varepsilon_k 
\end{cases}
\]

where \(\delta > 0\) is the constant maximum speed of the agents and \(\varepsilon_k > 0\) is a small design constant. In Equation (8.3), the switching term \(\beta_i(t)\) is introduced to avoid chattering due to small but acceptable errors in the desired interagent distances.

### 8.5.1.2 Control Law for One-DOF Agents

Let agent \(A_i\) have one DOF and \(A_j\) be the agent it follows. First, observe that once \(A_i\) satisfies its distance constraint with \(A_j\), it is free to move on the circle with center \(p_j\) and radius \(d_{ij}\) provided that it does not need to use the whole of its velocity capacity to satisfy \(|p_i - p_j| = d_{ij}\). Based on this observation and Guidelines G1 to G3, we design the following control scheme for \(A_i\):

\[
v_i(t) = \beta_i(t)v_{i1}(t) + \sqrt{1 - \beta_i(t)^2}v_{i2}(t)
\]

\[
\delta_{ji}(t) = (\delta_{jix}(t), \delta_{jiy}(t)) = p_j(t) - p_i(t)
\]

\[
\bar{\delta}_{ji}(t) = \delta_{ji}(t) - d_{ij}\delta_{ji}(t)/|\delta_{ji}(t)|
\]

(8.4)

\[
\beta_i(t) = \begin{cases} 
0, & |\bar{\delta}_{ji}(t)| < \varepsilon_k \\
\frac{|\bar{\delta}_{ji}(t) - \varepsilon_k|}{\varepsilon_k}, & \varepsilon_k \leq |\bar{\delta}_{ji}(t)| < 2\varepsilon_k \\
1, & |\bar{\delta}_{ji}(t)| \geq 2\varepsilon_k 
\end{cases}
\]
where

\[ v_{i1}(t) = \delta \delta_{ji}(t)/|\delta_{ji}(t)| \] (8.5)

\[ v_{i2}(t) = \alpha \beta_{i}(t) \text{sgn}(\langle \delta_{if}(t), \delta_{ji}^{\perp}(t) \rangle) \delta_{ji}^{\perp}(t) \]

\[ \delta_{if}(t) = p_{if}(t) - p_{i}(t) \]

\[ \delta_{ji}^{\perp}(t) = (-\delta_{jix}(t), \delta_{jiy}(t))/|\delta_{ji}(t)| \]

\[ \beta_{i}(t) = \begin{cases} 
0, & |\delta_{if}(t)| < \varepsilon_f \\
\frac{\|\delta_{if}(t)\| - \varepsilon_f}{\varepsilon_f} \varepsilon_f \leq |\delta_{if}(t)| < 2 \varepsilon_f \\
1, & |\delta_{if}(t)| \geq 2 \varepsilon_f 
\end{cases} \] (8.6)

\( \varepsilon_k, \varepsilon_f > 0 \) are small design constants and \( \langle \cdot, \cdot \rangle \) denotes the dot product operation. In Equation (8.4), via the switching term \( \beta_{i}(t) \), the controller switches between the translational action \( v_{i1} \) (given in Equation [8.5]) to satisfy \( |p_{i} - p_{j}| \equiv d_{ij} \) and the rotational action \( v_{i2} \) (given in Equation [8.6]) to move the agent \( A_i \) towards \( p_{if} \), which can take place only when \( |p_{i} - p_{j}| \) is sufficiently close to \( d_{ij} \).

In Equation (8.6), \( \delta_{ji}^{\perp}(t) \) is the unit vector perpendicular to the distance vector \( \delta_{ji}(t) = p_{j}(t) - p_{i}(t) \) with clockwise orientation with respect to the circle \( C(p_{j}(t), d_{ij}) \), and the term

\[ \text{sgn}(\langle \delta_{if}(t), \delta_{ji}^{\perp}(t) \rangle) \]

determines the orientation of motion that would move \( A_i \) towards \( p_{if} \). The switching term \( \beta_{i}(t) \) is for avoiding chattering due to small but acceptable errors in the final position of \( A_i \).

### 8.5.1.3 Control Law for Two-DOF Agents

If a given agent \( A_i \) has two DOFs (which can only happen if \( A_i \) is the leader of an acyclically led formation in our case), as it does not have any constraint to satisfy, it can use its full velocity capacity only to move towards its desired final position \( p_{if} \). Hence the velocity input at each time \( t \) can be simply designed as a vector with magnitude \( \delta \) in the direction of \( p_{if} - p_{i}(t) \):

\[ v_{i}(t) = \delta \beta_{i}(t) \delta_{if}(t)/|\delta_{if}(t)| \]

\[ \delta_{if}(t) = p_{if} - p_{i}(t) \]

\[ \beta_{i}(t) = \begin{cases} 
0, & |\delta_{if}(t)| < \varepsilon_f \\
\frac{\|\delta_{if}(t)\| - \varepsilon_f}{\varepsilon_f} \varepsilon_f \leq |\delta_{if}(t)| < 2 \varepsilon_f \\
1, & |\delta_{if}(t)| \geq 2 \varepsilon_f 
\end{cases} \] (8.7)

The switching term \( \beta_{i}(t) \) again prevents chattering due to small but acceptable errors in the final position of \( A_i \).
8.5.2 Stability and Convergence

In this section we informally discuss the stability and convergence properties associated with the application of the control laws designed in Section 8.5.1 to each of the classes of acyclically led and cyclically led persistent formations separately.

8.5.2.1 Acyclically Led Minimally Persistent Formations

Consider Problem 1 for an acyclically led minimally persistent formation $F$ with $m \geq 3$ agents $A_1, \ldots, A_m$, where without loss of generality, $A_1$ is the leader, $A_2$ is the first follower, and the other agents are ordinary followers (such a formation is depicted in Figure 8.4). Note here that, by the assumption at the end of Section 8.4, $A_2$ follows $A_1$. In this case based on the proposed control scheme in Section 8.5.1 $A_1$ uses the control law (8.7); $A_2$ uses the control law (8.4) to (8.6); each of $A_3, \ldots, A_m$ uses the control law (8.3).

Following is an informal sketch of a possible analysis of the stability and convergence properties of $F$ during its motion, noting that the formal complete analysis was not completed during the submission of this chapter. Consider dynamic behavior of each agent separately. The leader agent $A_1$ uses the control law (8.7). Hence, defining the Lyapunov function $V_1(t) = \frac{1}{2} \delta_{1f}^T(t) \delta_{1f}(t)$, from Equation (8.7) it follows that:

$$
\dot{V}_1(t) = -\delta_{1f}^T(t) v_1(t) \\
= -\delta \dot{P}_1(t) \delta_{1f}^T(t) \delta_{1f}(t) / |\delta_{1f}(t)| \\
= -\delta \dot{P}_1(t) |\delta_{1f}(t)| \\
= \begin{cases} 
0, & \text{if } |\delta_{1f}(t)| < \varepsilon_f \\
-\delta |\delta_{1f}(t)| \frac{|\delta_{1f}(t)| - \varepsilon_f}{\varepsilon_f} \leq -\delta \left(|\delta_{1f}(t)| - \varepsilon_f\right), & \text{if } \varepsilon_f \leq |\delta_{1f}(t)| < 2\varepsilon_f \quad (8.8) \\
-\delta |\delta_{1f}(t)| \leq -2\delta \varepsilon_f, & \text{if } |\delta_{1f}(t)| \geq 2\varepsilon_f
\end{cases}
$$
Therefore, we have $|\delta_1(t)| \leq |\delta_1(0)|; \forall t \geq 0$ and $\lim_{t \to \infty} |\delta_1(t)| \leq \varepsilon_f$, that is, $p_1(t)$ is always bounded and asymptotically converges to the ball $B(p_{1f}, \varepsilon_f)$, that is, the ball with center $p_{1f}$ and radius $\varepsilon_f$. It can be further deduced from Equation (8.8) that $p_1(t)$ enters the ball $B(p_{1f}, 2\varepsilon_f)$ in finite time and remains there.

A similar but relatively longer analysis can be done for the dynamic behavior of $A_2$ defining the Lyapunov functions

$$V_{21}(t) = \frac{1}{2} \delta_{12}^T(t) \delta_{12}(t), \quad V_{2f}(t) = \frac{1}{2} \delta_{2f}^T(t) \delta_{2f}(t)$$

and examining $\dot{V}_{21}(t)$ and $\dot{V}_{2f}(t)$ together with $\dot{V}_1(t)$ and the control law Equations (8.4) to (8.6). This analysis is expected to establish the conditions under which $p_2(t)$ remains bounded, and converges to finite time-varying balls around $p_1(t)$ for $t \geq 0$ with certain radii as well as a fixed ball around $p_{2f}$ with a certain radius.

Similarly, for agents $A_i$ where $i \in \{3, 4, \ldots, m\}$, analyzing $\dot{V}_{id}(t)$ for

$$V_{id}(t) = \frac{1}{2} \delta_{id}^T(t) \delta_{id}(t)$$

it appears possible that boundedness and convergence properties of each $p_i(t)$ can be established. Combining this result with the above ones for agents $A_1$, $A_2$ and via some geometric analysis on the definition of $p_{id}$ in (Equation 8.2), the conditions to guarantee convergence of all the agents to their final desired positions, as well as satisfaction of the distance constraints within certain error tolerance bounds, can be deduced.

Note here that the discussions above, as well as the applicability of the control laws for agents $A_i$ ($i \in \{3, 4, \ldots, m\}$), are valid if and only if $p_{id}(t)$ in Equation (8.2) is well defined, that is, the circles $C(p_j(t), d_{ij})$ and $C(p_k(t), d_{ik})$ intersect for all $t \geq 0$. Via a geometric analysis of accumulation of the position errors it is observed that Equation (8.2) can be guaranteed to be well defined selecting the constant $\varepsilon_k$ sufficiently small.

Some simulation results and discussions on testing of the control structure described above on acyclically led persistent formations can be found in Reference [25]. The results shown in Reference [25] for a four-agent example formation indicate that the control goals are successfully achieved where each agent satisfies its distance constraints all the time during motion with a significantly small error (less than 2% of the distance to be kept).

### 8.5.2.2 Cyclically Led Minimally Persistent Formations

Consider a cyclically led minimally persistent formation $F$ with $m \geq 3$ agents $A_1, \ldots, A_m$, where $A_1$, $A_2$, and $A_3$ are the coleaders, and the other agents are ordinary followers (such a formation is depicted in Figure 8.5). By the assumption at the end of Section 8.4 which involves some loss of generality, assume also that $A_2$ follows $A_1$, $A_3$ follows $A_2$, and $A_1$ follows $A_3$. In this case, based on the proposed control scheme in Section 8.5.1, each of $A_1$, $A_2$, and $A_3$ uses the control law (8.4) to (8.6) and each of $A_4, \ldots, A_m$ uses the control law (8.3).
For the remaining agents it appears possible that boundedness and convergence properties of each $p_i(t)$ can be established in a similar manner to that suggested for acyclically led formations. Combining all these results, the conditions guaranteeing cohesive motion as well as stability and convergence of the entire formation to the desired destination can be deduced.

Again, the formal analysis corresponding to the above sketch has not been completed yet. Nevertheless, the corresponding simulation results and discussions of Reference [25] on testing of the control structure described in Section 8.5.1 demonstrate that the global stability and convergence properties or cyclically led persistent formations are comparable to those of the acyclically led formations. One distinction observed in the results shown in Reference [25] is that the agents follow longer and more indirect paths than the acyclically led case. This is mainly because of guidance of the whole formation by a coleader triple which make constrained motions as described by Equations (8.4) to (8.6).

8.5.3 More Complex Agent Models

In this chapter, we have designed control schemes for and analyzed the problem of cohesive motion of persistent formations based on the velocity integrator model (8.1) for the agent kinematics. The actual kinematic or dynamic model of agents in a real-life formation, however, would be more complex than a velocity integrator in general. Therefore, a more practical design and analysis procedure for the cohesive motion problem would require a more complex agent model than Equation (8.1). The form of such a model for a particular application, of course, will depend on the specifications of the agents used in this application.

Some of the widely used continuous-time agent models used in the formation control literature corresponding to the practical experimental agents

\footnote{Discrete-time counterparts of these models exist and are used in the literature as well.}
(e.g., robots or vehicles) of interest are the double integrator or point mass dynamics model, where the acceleration term \( \dot{v}_i = a_i \) is added to the model (8.1) and it is assumed that the control input is the vectorial acceleration \( \bar{a}_i \) [21,22]; the fully actuated uncertain dynamics model [11,13]:

\[
M_i(p_i) \ddot{p}_i + \eta_i(p_i, \dot{p}_i) = u_i
\]

where \( M_i \) represents the mass or inertia matrix, \( \eta_i \) represents the centripetal, Coriolis, gravitational effects, and other additive disturbances, and \( u_i \) is the control signal that can be applied in the form of a vectorial force; and the nonholonomic unicycle dynamics model [19]:

\[
\begin{align*}
(\dot{x}_i, \dot{y}_i) &= (v_i \cos \theta_i, v_i \sin \theta_i) \\
\dot{\theta}_i &= \omega_i \\
\dot{v}_i &= \frac{1}{m_i}u_{i1} \\
\dot{\omega}_i &= \frac{1}{v_i}u_{i2}
\end{align*}
\]  

(8.9)

where \( p_i(t) = (x_i(t), y_i(t)) \) denotes the position of \( A_i \) as before and \( \theta_i(t) \) denotes the orientation or steering angle of the agent with respect to a certain fixed axis, \( v_i \) is the translational speed, \( \omega_i \) is the angular velocity, and the control input signals \( u_{i1}, u_{i2} \) are, respectively, the force and torque inputs.

A simplified form of Equation (8.9) is the nonholonomic unicycle kinematics model [9]:

\[
(\dot{x}_i, \dot{y}_i) = (v_i \cos \theta_i, v_i \sin \theta_i) \\
\dot{\theta}_i = \omega_i
\]  

(8.10)

where it is assumed that the translational speed \( v_i \) and the angular velocity \( \omega_i \) can be applied as control inputs directly.

Some preliminary results, based mainly on simulation studies, on the solution of Problem 1 for the nonholonomic unicycle kinematic agent model (8.10) are presented in Reference [25]. The control schemes used in this work employ a so-called “separation-separation control” idea Reference [9], which was originally developed for following a mobile robot by another one at a specified distance or relative position, and are not direct extensions of the designs presented in Section 8.5.1. Nevertheless, in the simulations in Reference [25] using the new control scheme, it is observed that the control goal in Problem 1 is achieved, although the performance is poor compared to the performance for the velocity integrator model in terms of both the path length and the distance constraints.

The simulation results in Reference [25] demonstrate that cohesive motion control with agent kinematics or dynamics models that are more complex than the velocity integrator model is feasible. Design of an enhanced control scheme to obtain a better performance for the unicycle kinematic agent model (8.10), as well as similar designs for the double integrator and fully actuated uncertain dynamics models, are currently being investigated by the authors.
8.6 Discussions and Future Directions

In this chapter we have analyzed persistent autonomous multiagent formations based on a recently developed theoretical framework of graph rigidity and persistence. We have reviewed the general characteristics of rigid and persistent graphs and their implications on the control of persistent formations, and presented some operational criteria to check the persistence of a given formation. Using these characteristics and criteria, we have analyzed certain persistence acquisition and maintenance tasks. Later, we have analyzed cohesive motion of persistent autonomous formations and presented a set of distributed control schemes to cohesively move a given persistent formation with specified initial position and orientation to arbitrary desired final position and orientation.

There still exist open problems or designs to be completed related to discussions presented on each of characteristics, persistence acquisition, persistence maintenance, and cohesive motion of persistent formations. Relevant to the studies presented in Sections 8.2 and 8.3, the authors are currently 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 aerial vehicles. Beside these, the general forms of splitting and closing rank problems for persistent formations remain open, as well as the general solution to persistence acquisition problems defined in Section 8.3.1.

Related to the cohesive motion problem, the authors are currently working on analyzing and enhancing the control laws and strategies presented for optimality in terms of, for example, the total displacement of all agents; design of similar control schemes for more complex agent models discussed in Section 8.5.3; and solution of the cohesive motion problem in the existence of obstacles in the region of interest. Different approaches to these ongoing studies as well as consideration of various real-life effects such as distance measurement noises, lack of global position sensing for some agents, and so on, may constitute different future research directions.

Acknowledgment

The work of B. Fidan, B. Anderson, and C. Yu 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. J. Hendrickx holds an FNRS fellowship (Belgian Fund for Scientific Research).
His work is supported by the Belgian Programme on Interuniversity Attraction Poles initiated by the Belgian Federal Science Policy Office, and The Concerted Research Action (ARC) "Large Graphs and Networks" of the French Community of Belgium. The scientific responsibility rests with its authors.

References


