17th Mediterranean Conference on Control & Automation
Makedonia Palace, Thessaloniki, Greece
June 24 - 26, 2009

Edge Contraction Based Maintenance of Rigidity in Multi-Agent Formations During Agent Loss

Barış Fidan[1], Julien M. Hendrickx[2], and Brian D.O. Anderson[1]
[1]National ICT Australia (NICTA) and The Australian National University – Research School of Information Sciences & Engineering, Canberra, Australia
[2]Laboratory for Information and Decision Systems, Massachusetts Institute of Technology, Cambridge, USA and Department of Mathematical Engineering, Université catholique de Louvain, Louvain-la-Neuve, Belgium

***Abstract*—This paper proposes a systematic approach to the problem of restoring rigidity after loss of an agent, for two-dimensional rigid multi-agent formations based on a particular graph operation, the edge contraction operation. A rigidity maintenance method is proposed, for the cases where an agent is lost in an arbitrary two-dimensional rigid formation, to restore rigidity by transferring all links to which this agent was incident on to one of its neighbors. From a graph theoretical point of view, this corresponds to contraction of a certain edge incident to the vertex representing the agent being lost.**

I. Introduction

Distributed formation control of autonomous multi-agent systems has gained significant attention in the last decade in parallel with the interest in the practical applications of such systems involving teams of unmanned aerial and ground vehicles, combat and surveillance robots, underwater vehicles, wireless sensor networks, etc [1]–[5]. For various reasons, multi-agent systems, where each agent can be a vehicle, a robot, or a sensor unit, constitute a more effective sensor than a single vehicle in many cases [6]. One example motivator from surveillance applications is that multiple mobile sensors can cover a region of interest more quickly than a single sensor, even with limited individual sensing range. Other examples in different application areas can be seen in [6].

In most of the multi-agent system applications mentioned above, accurate knowledge and control of the agents' relative positions in the formation is essential, where we use the term *formation* for a collection of agents moving in real 2- or 3-dimensional space to fulfill certain mission requirements. Correspondingly, many multi-agent system applications require the shape of the corresponding formation to be preserved. For example, target localization by a group of unmanned airborne vehicles (UAVs) using either angle of arrival data or time difference of arrival information is best achieved (in the sense of minimizing localization error) when the UAVs are located at the vertices of a regular polygon [7], [8]. This objective can be achieved by explicitly keeping some inter-agent distances constant. In other words, some inter-agent distances are explicitly maintained constant so that all the inter-agent distances remain constant. The information structure arising from such a system can be efficiently modeled using graph theoretical notions as detailed below.

In this paper, we consider the motion of the entire formation rather than individual agent behaviors and assume a point-agent system model [1], [9], noting that 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 [1] and the references therein. We represent each multi-agent 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, i.e., 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 moves as a cohesive whole. Typically the agent pairs in F whose inter-distances are explicitly maintained are the ones having information (i.e., sensing and communication) 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) at a desired value d_{ij} with an information link in between the two corresponds to keeping the length of the edge $(i, j) \in E_F$ constant at d_{ij}. A rigid formation is further called *minimally rigid*, if minimum possible number of links are used to maintain the rigidity of the formation.

In many autonomous multi-agent formation applications, 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 [1]. In [3] three key categories of such operations on rigid formations have been analyzed, with focus on preservation of rigidity during these operations: *Merging*, *splitting* and *closing ranks*.

The *merging* operation is out of the scope of this paper, and the details can be found in [3]. In *splitting*, the case is considered where a rigid formation F with underlying graph $G_F = (V_F, E_F)$ is split into two formations F_1, F_2 with underlying graphs $G_{F1} = (V_{F1}, E_{F1})$, $G_{F2} = (V_{F2}, E_{F2})$, respectively, (where $V_F = V_{F1} \cup V_{F2}$) 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 edges to E_{F1}, E_{F2}) such that both F_1 and F_2 become rigid.

The operation focused in this paper, the *closing ranks* operation can be thought 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 rigid formation, and the *closing ranks problem* is to establish new links between certain pairs among the remaining agents such that the new formation (formed after establishment of the new links) is rigid as well. Note here that splitting can be thought as a generalized closing ranks operation (defined for the loss of a set of agents instead of a single agent), observing that the scenario of the above splitting problem for, e.g., the post-split formation F_1 can be equivalently reformulated as F_1 being what is left when F, having initially F_2 as its sub-formation, then loses the agents in the sub-formation F_2. This observation is useful in treating splitting problems using certain results derived for the closing ranks problem, including the results of this paper.

The *closing ranks problem* is approached in two different ways in [3], [10]. In [3], based on the graph theoretic results of [11], it is shown that, in the case of loss of an agent from a minimally rigid formation, there exist new links between the neighbors of the lost agent such that insertion of those links results in preservation of minimal rigidity. Although [3] provides existence results and the minimal number of edges needed to maintain minimal rigidity, it does not provide a closing ranks algorithm that can be implemented using only local information. [10] introduces two closing ranks algorithms, named *double patch* and *wheel patch*, which can be implemented using only local information (i.e., knowing only the set of neighbors of the lost agent), although these two algorithms require a non-minimal number of new links.

In this paper, we propose a new systematic approach to the closing ranks problem based on a particular graph operation, the *edge contraction* operation, whose details are given in Section III. We prove that when an agent is lost in an arbitrary two-dimensional rigid formation, rigidity can always be restored by transferring all links to which this agent was incident to one of its neighbors. From a graph theoretical point of view, this corresponds to contraction of a certain edge incident to the vertex representing the agent being lost.

II. Rigid Formations and Closing Ranks

In this section, we give formal definitions of the *rigidity* and *closing ranks* notions and present a brief review of the fundamental characteristics of rigid formations and available results on the closing ranks problem to the extent needed for the analysis in the following sections. For details the reader may refer to [3], [9], [12], [13].

In the paper we focus on formations in $\Re^2$ (2-dimensional Euclidean space), noting that discussions on extension of the results to be presented in the following sections to $\Re^3$ can be found in [14], extended version of this paper. Hence, the *rigidity* and *closing ranks* notions and characteristics are introduced in $\Re^2$ in this section, although most of them can be generalized for arbitrary dimensional space $\Re^d$ $(d \in \{2, 3, \dots\})$ [9].

A. Rigidity of Multi-Agent Formations

We formally call a formation F in $\Re^2$ *rigid* if its underlying graph G_F is *generically* 2*-rigid*, where *generic* 2*-rigidity* of a graph is defined in the sequel. In $\Re^2$, a *representation* of a graph $G = (V, E)$ with vertex set V and edge set E is a function $p : V \to \Re^2$, associating a *position* p_i to each vertex $i \in V$. The graph-representation pair (G, p) above is called a 2-dimensional *framework*. A framework (G, p) is *rigid* if for any deformation $\tilde{p} : V^+ : (i, t) \to \tilde{p}_i(t)$, continuous with respect to t, and such that $\tilde{p}_i(0) = p_i$ for all i, if $\|\tilde{p}_i(t) - \tilde{p}_j(t)\| = \|p_i - p_j\|$ holds for all t and every edge $(i, j) \in E$, then it also holds for all t and for every pair of vertices $i, j \in V$, where $\| \cdot \|$ denotes the Euclidean norm. In other words a framework is rigid if any continuous deformation that preserves the "length" of its edges also preserves the distance between every two agents. A graph is said to be *generically* 2*-rigid* or simply 2*-rigid* if almost all its frameworks in $\Re^2$ are rigid. Some discussions on the need for using the qualifiers "generic" and "almost all" can be found in [11], [15]. One reason for using these terms is to avoid the problems that may arise when three or more vertices have collinear positions.

An important notion in rigidity analysis is *minimal rigidity*. A graph G is called *minimally* 2*-rigid* if it is 2-rigid and has no 2-rigid subgraph with the same set of vertices as G and a smaller number of edges than G. Provably equivalently, a graph is *minimally* 2*-rigid* if it is 2-rigid and if no single edge can be removed without losing 2-rigidity. Fundamental characteristics of rigid and minimally rigid graphs and some of their applications in autonomous formation control can be found in [3], [11]–[13]. Theorem 1 and Lemmas , 1, 2 below are three key results related to these characteristics.

Theorem 1: Any graph $G = (V, E)$ with at least 2 vertices is 2-rigid if and only if there exists a subset $E' \subseteq E$ of edges such that the graph $G' = (V, E')$ is minimally 2-rigid and satisfies the following:

(i) $|E'| = 2|V| - 3$.
(ii) Any subgraph $G'' = (V'', E'')$ of G' with at least n vertices satisfies $|E''| \leq 2|V''| - 3$.

Two operations on rigid graphs play a central role in the results below: *vertex addition* and *edge splitting*. Let G be a 2-rigid graph. *Vertex addition* consists of adding a vertex k to G, and connecting it to 2 distinct vertices of G; *edge splitting* consists of removing an edge (v, w) from G, adding a vertex k, and connecting it to v, w and one other distinct vertex of G. Both operations, see Figure 1, preserve rigidity, indeed minimal rigidity:

Lemma 1: Let G be a 2-rigid graph. Every graph G' obtained by performing a vertex addition or an edge splitting operation on G is 2-rigid. If G is minimally 2-rigid then G' is minimally 2-rigid.

The next lemma is a form of converse. Both together can show that every minimally rigid graph can be constructed by sequential vertex additions and edge splittings.

Lemma 2: Let k be a vertex in a 2-rigid graph G. If k has degree 2, then the graph G_{-k} obtained by removing k and its incident edges, from which G can be obtained by performing a vertex addition operation, is 2-rigid. If k has degree 3, then there exists a 2-rigid graph G'_{-k} from which G can be obtained by adding k using an edge splitting operation, and at least two of the neighbor vertices of k in G are adjacent in G'_{-k}. If G is minimally 2-rigid then G_{-k} and G'_{-k} are minimally 2-rigid.

Based on Lemma 2, we can define the minimal rigidity preserving operations *reverse vertex addition* and *reverse edge splitting*: For a minimally 2-rigid graph G having at least one vertex with degree 2, the *reverse vertex addition* operation consists of deleting a 2-degree vertex k and its incident edges from G. For a minimally 2-rigid graph $G = (V, E)$ having at least one vertex with degree 3, the *reverse edge splitting* operation consists of deleting a 3-degree vertex k that is originally adjacent to vertices $h, i, j \in V$, and adding one edge among (h, i), (i, j), and (h, j) in such a way that the graph obtained is minimally 2-rigid. Note that the *reverse edge splitting* operation can be performed on every vertex with degree 3 in a minimally rigid graph [11], [15], but one cannot freely choose the edge to be added [15]. Examples of the *vertex addition*,*edge splitting*, *reverse vertex addition*, and *reverse edge splitting* operations can be seen in Figure 1.

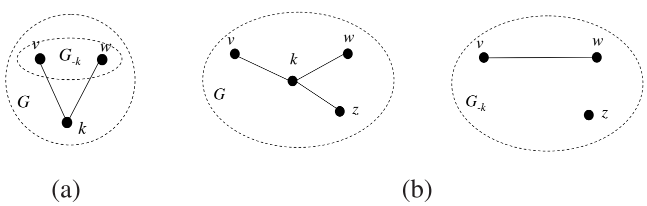

Figure 1. (a) Vertex addition and reverse vertex addition, and (b) edge splitting and reverse edge splitting.

B. Closing Ranks Problem

As introduced in Section I, the *closing ranks problem* deals with the addition of links to a rigid formation that is "damaged" by losing one of its agents, in order to regain its rigidity. The *rigidity preserving closing ranks problem* for a 2-dimensional formation F can be formally defined in terms of its underlying graph G_F as follows: Suppose that $G_F = (V_F, E_F)$ is 2-rigid, and let $G'_F = (V_F \setminus \{v_r\}, E_F \setminus E_r)$ be a graph obtained from G_F by removing a vertex $v_r \in V_F$ and all the edges in E_F incident to v_r. The *closing ranks problem* is the problem of finding a set E_n of edges that, if added to G'_F, would render it 2-rigid, i.e. finding E_n such that $G_{Fn} = (V_F \setminus \{v_r\}, (E_F \setminus E_r) \cup E_n)$ is $2-rigid$. The *minimal rigidity preserving closing ranks problem* can be defined similarly, replacing each "2-rigid" in the above definition with "minimally 2-rigid". In [3], based on [11], the following existence and necessary lower bound results are established for the (number of) new edges to be added in *minimal rigidity preserving closing ranks problem*:

Theorem 2: [3] Assume that a vertex v of degree k and its incident edges are removed from a minimally 2-rigid graph, where $k \geq 2$. To regain minimal 2-rigidity, $k - 2$ new edges (or no edges if $k = 2$) should be inserted and those new edges can be inserted between only the neighbors of v, without using any other vertex as an end-vertex of those inserted edges.

Although Theorem 2 provides existence results and the minimal number of edges needed to maintain minimal rigidity, it does not tell which edges are to be added among the neighbors of the removed vertex. Implementation of an algorithm purely based on Theorem 2 would require a search over all possible sets of $k - 2$ edges between neighbors of the removed vertex v for checking minimal rigidity of the graph obtained by insertion of each of these edge sets. The results of the next section provide a more systematic and computationally more effective *closing ranks* algorithm, yet requiring a number of new edges (links) that is either equal to or only one more than the lower bound indicated in Theorem 2.

III. Edge Contraction For Closing Ranks

In the graph theory literature, *edge contraction* is defined as the operation by which two adjacent vertices $v_1, v_2 \in V$ of a graph $G = (V, E)$ are merged into one vertex, which is adjacent with all the vertices in V that v_1 and v_2 were adjacent to, and the edge $e = (v_1, v_2) \in E$ is removed [16]. The merged vertex and the resultant graph are denoted, respectively, by $v_1 \setminus v_2$ and $G \setminus e$. Above, for each vertex $v_3 \in V$ to which v_1 and v_2 are both adjacent in G, $G \setminus e$ will have a double edge between $v_1 \setminus v_2$ and v_3. In our study, we do not allow double edges in a graph, hence re-define *edge contraction* as the above operation plus merging of any double edges between $v_1 \setminus v_2$ and another vertex v_3 in a single edge $(v_1 \setminus v_2, v_3)$. In the rest of the paper, we use this re-defined form of *edge contraction*, and to make distinction from the more widely used form of it (allowing double edges) in the literature, for a given graph $G = (V, E)$ and an edge $e \in E$, we denote the graph obtained from G by (re-defined) contraction of e as $G_{c(e)}$. We say that an edge $e \in E$ in a 2-rigid graph $G = (V, E)$ is 2-*contractible* if $G_{c(e)}$ is also 2-rigid (see Figure 2 for examples).

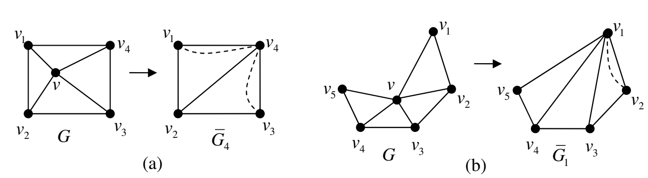

Figure 2. Edge contraction of rigid graphs with application to closing ranks of rigid formations: Starting with the 2-rigid graph G, $\bar{G}_4 = G_{c((v,v_4))}$ is obtained by contracting the edge (v, v_4) in Case (a) and $\bar{G}_1 = G_{c((v,v_1))}$ is obtained by contracting (v, v_1) in Case (b), which are both 2-rigid graphs as well. Each case corresponds to a closing ranks scenario, where a rigid formation with underlying graph G loses an agent represented by v and it is desired to maintain the rigidity by establishing new links among the neighbors of the lost agent.Dashed lines indicate deleted double edges.

Remark 1: For a given graph $G = (V, E)$ and an edge $e \in E$, the number of edges in $G_{c(e)}$ is always less than $|E|$.

In this section, our aim is to prove the following proposition, which is later used to form a basis for establishing a class of algorithms to solve closing ranks problems:

Proposition 1: Consider a 2-rigid graph $G = (V, E)$ and a vertex $v \in V$ with incident vertices $v_1, \ldots, v_m$. There exists $i \in \{1, \ldots, m\}$ such that $\bar{G}_i \triangleq G_{c((v,v_i))}$ is 2-rigid.

A stronger form of this proposition, Theorem 4, is proven in the sequel, i.e., Proposition 1 is an immediate corollary of Theorem 4 below. In proving Theorem 4, we use the following lemmas, whose proofs are omitted here due to space limitations and can be found in [14]:

Lemma 3: Let $G(V, E)$ be a minimally 2-rigid graph in which no vertex has degree 2. Then, there is no vertex in V that is incident to all vertices with degree 3.

Lemma 4: Let k be a vertex of degree 2 in a minimally 2-rigid graph G, and G_{-k} be the minimally rigid graph obtained by removal of k applying a reverse vertex addition. Let v, w be the neighbors of k in G. A 2-contractible edge of G_{-k} that is incident to at most one vertex among $\{v, w\}$ is also 2-contractible in G.

Lemma 5: Let k be a vertex of degree 3 in a minimally 2-rigid graph G, and G_{-k} be the minimally 2-rigid graph obtained by removal of k applying a reverse edge splitting. Let v, w, z be the neighbors of k in G. A 2-contractible edge of G_{-k} that is incident to at most one vertex among $\{v, w, z\}$ is also 2-contractible in G.

Theorem 3: Let $G = (V, E)$ be a minimally 2-rigid graph with at least three vertices. For any vertex $i \in V$, there exist $j_1, j_2 \in V$, neighbors of i, such that both $G_{c((i,j_1))}$ and $G_{c((i,j_2))}$ are rigid. In other words, all vertices in V are incident to at least two 2-contractible edges in E.

Proof: We prove the theorem by induction on $|V|$, the number of vertices: The result obviously holds for $|V| = 3$. Suppose, for proof by induction, that it holds for all minimally 2-rigid graphs with $|V| = n - 1$ vertices for a given integer n. Below, we prove that it then also holds for any arbitrary minimally 2-rigid graph $G = (V, E)$ with $|V| = n$ for the following two cases separately: (1) G contains a vertex k of degree 2 and (2) G contains no vertex of degree 2.

Case (1) G contains a vertex k of degree 2: Let v, w be the neighbors of k. In this case, by Lemma 1, the graph G_{-k} obtained by performing a reverse vertex addition, that is removing k and its incident edges, is minimally 2-rigid (see Figure 1(a)). We consider three subcases:

Subcase (1.a) $i = k$: The contraction of (k, v) or of (k, w) leads to a graph containing G_{-k} as a subset (actually, to G_{-k} plus an edge (v, w) if it is not already present in G_{-k}) and which is therefore rigid. Hence, $i = k$ is incident to two 2-contractible edges (k, v) and (k, w).

Subcase (1.b) $i = v$ or $i = w$: Without loss of generality assume that $i = v$. By the induction hypothesis, in G_{-k}, v is incident to at least one 2-contractible edge different from (v, w), which, by Lemma 4, is also 2-contractible in G. Moreover, we know from Subcase (1.a) that (v, k) is 2-contractible G. Hence v is incident to two 2-contractible edges.

Subcase (1.c) $i \notin \{k, v, w\}$: By the induction hypothesis, there exist two distinct 2-contractible edges (i, j_1) and (i, j_2) in G_{-k}, and each of them is adjacent to at most one vertex among v and w since $i \notin \{k, v, w\}$. Lemma 4 implies then that they are also 2-contractible in G.

Case (2) G contains no vertex of degree 2: Since the average degree in a minimally 2-rigid graph is smaller than 4, it must contain vertices of degree 3. Let k be such a vertex of degree 3 with neighbor vertices v, w, z. Lemma 2 implies that there exists a minimally 2-rigid graph G_{-k} that can be obtained from G performing a reverse edge splitting operation removing k and adding an edge (v, w) between two vertices v, w that are not adjacent to each other in G but both adjacent to k (see Figure 1(b)). By the induction hypothesis, every vertex of G_{-k} is incident to at least two 2-contractible edges; and by Lemma 5, every 2-contractible edge of G_{-k} which is incident to less than two vertices among $\{v, w, z\}$ is also 2-contractible in G. Therefore, all vertices of G different from $\{v, w, z, k\}$ are incident to at least two 2-contractible edges.

Consider z, and a vertex k' of degree 3 that is not incident to z. The existence of such a vertex is guaranteed by Lemma 3. By the induction hypothesis, z is incident to two 2-contractible edges in $G_{-k'}$, the graph obtained from G by removing k' using a reverse edge splitting. Each of these two edges is incident to at most one neighbor of k' in G, as z is not a neighbor of k'. Therefore, by Lemma 5, they are also 2-contractible in G, so that z is incident to two 2-contractible edges in G. The same argument applies to v and w, hence it remains to prove that k is incident to two 2-contractible edges to complete the proof: Since each of $G_{c((k,v))}$ and $G_{c((k,w))}$ contains G_{-k} as a subgraph (see Figure 3) and hence is 2-rigid, (k, v) and (k, w) are 2-contractible. Therefore, k is incident to two 2-contractible edges. ■

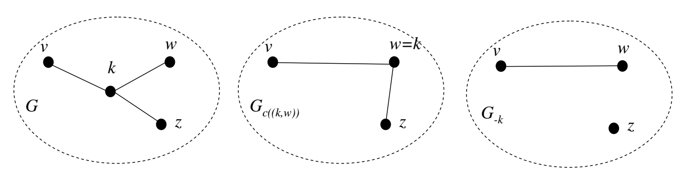

Figure 3. Contraction of (k, w) in the minimally rigid graph G: $G_{c((k,w))} \supseteq G_{-k}$.

Theorem 4: Let i be a vertex of a rigid graph $G = (V, E)$ with at least three vertices. There exist j_1, j_2, neighbors of i, such that both $G_{c((i,j_1))}$ and $G_{c((i,j_2))}$ are 2-rigid. In other words, all vertices in V are incident to at least two 2-contractible edges in E.

Proof: Let $G' = (V, E')$ be a minimally 2-rigid subgraph of G having the same vertex set V. Existence of such a subgraph is guaranteed by Theorem 1. By Theorem 3, there exist j_1, j_2, neighbors of i, such that both $G'_{c((i,j_1))}$ and $G'_{c((i,j_2))}$ are 2-rigid. Consider the graphs $G_{c((i,j_1))}$ and $G_{c((i,j_2))}$. They satisfy $G'_{c((i,j_1))} \subseteq G_{c((i,j_1))}$ and $G'_{c((i,j_2))} \subseteq G_{c((i,j_2))}$. Hence, since $G'_{c((i,j_1))}$ and $G'_{c((i,j_2))}$ are 2-rigid, both $G_{c((i,j_1))}$

and $G_{c((i,j_2))}$ are 2-rigid, which completes the proof. ■

Remark 2: In a minimally 2-rigid graph $G = (V, E)$, there does not necessarily exist an edge $e \in E$ such that $G_{c(e)}$ is also minimally 2-rigid, as demonstrated in Figure 4.

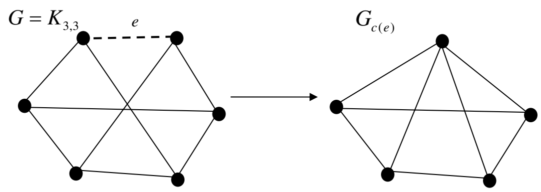

Figure 4. Edge contraction in $G = K_{3,3}$: G is minimally 2-rigid; however $G_{c(e)}$ is non-minimally 2-rigid for any edge e, noting that all edges in G are topologically equivalent.

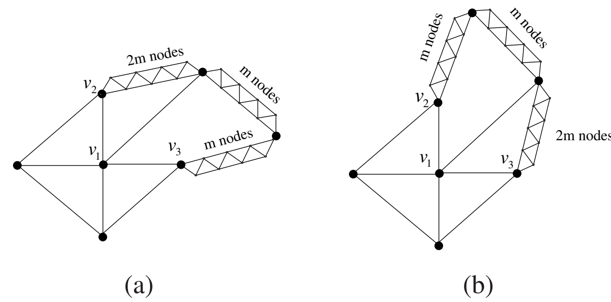

Figure 5. Illustration of the fact that knowing whether or not an edge is 2-contractible may require the knowledge of the structure of the graph at an arbitrarily large distance from the edge considered. In the minimally rigid graph (a), the edge (v_1, v_2) is not 2-contractible, while the edge (v_1, v_3) is 2-contractible. In (b), (v_1, v_3) is 2-contractible and (v_1, v_2) is not. The two graphs can however not be distinguished without exploring vertices and edges at a hop-distance of m (where m can be arbitrarily large) from the vertex v_1.

IV. On Designing A Systematic Closing Ranks Algorithm

In a *minimal rigidity preserving closing ranks problem*, if one agent of degree d_v (i.e., having d_v neighbors) is lost in a minimally rigid formation F with underlying graph $G_F = (V_F, E_F)$, where $v \in V_F$ represents the lost agent, Theorem 2 implies that we just need to add $d_v - 2$ edges between the (former) neighbors of v, without telling which $d_v - 2$ edges are to be added. A pebble game algorithm might be used to choose $d_v - 2$ edges among the $d_v(d_v - 1)/2$ possible. Using the results of Section III, however, to solve the closing ranks problem in case of loss of an agent in a 2-dimensional rigid formation F, that is represented by v in the underlying graph $G_F = (V_F, E_F)$, we just need to find a 2-contractible one among the d_v edges incident on v in G_F. With a crude approach, this search can be performed by checking the 2-rigidity of each of the d_v graphs $G_{Fc(e_1)}, \ldots, G_{Fc(e_{d_v})}$, where $\{e_1, \ldots, e_{d_v}\} \subset E_F$ is the set of edges incident on v. Moreover, several heuristic approaches can be used to discard some edges which cannot be contracted without losing rigidity. In this section, we discuss design of a more systematic closing ranks algorithm based on edge contraction, which would involve a better way of finding the 2-contractible edges.

A. Checking 2-Contractibility of An Arbitrary Edge

Unlike many recurrence proofs, our proofs of Theorems 3 and 4 are not constructive. As such, they do not allow one to just build a minimally rigid graph using vertex addition and edge splitting operations, and keep track of all contractible edges at each step. Further work needs to be dedicated to the development of efficient methods to determine whether an edge is 2-contractible or not, or to find the 2-contractible edges, leading to decentralized algorithms. It is not known yet whether a decentralized algorithm to check if an edge is 2-contractible can be designed. Such an algorithm would however, in some cases, need to use information on vertices and edges that can be at arbitrarily large distance from the edge considered, as shown by the example in Figure 5.

B. Special Cases

Despite the demonstrations of Section IV-A about potential requirements of massive information by closing ranks algorithms based on edge contraction, there exist classes of generic cases where one can easily determine that an edge is or is not 2-contractible, as implied by the following two propositions and their corollary, whose proofs are omitted here due to space limitations and can be found in [14]:

Proposition 2: Let v be a vertex in a minimally 2-rigid graph $G = (V, E)$, such that the subgraph $G_n = (V_n, E_n)$ containing only the neighbors of v (but not v) and the edges induced by these neighbor vertices in G is a tree (viz. connected acyclic graph). Then, performing an edge contraction on (v, w) leads to a minimally 2-rigid graph, where w is any leaf (viz. vertex of degree 1) of G_n.

Proposition 3: Let v be a vertex in a 2-rigid graph $G = (V, E)$, such that the subgraph $G_n = (V_n, E_n)$ containing only the neighbors of v (but not v) and the edges induced by these neighbor vertices in G is a tree. Then, edge contraction of (v, w) leads to a 2-rigid graph, where w is any leaf of G_n.

Corollary 1: Consider an arbitrary vertex v in a graph $G = (V, E)$. Let $V_n = \{w_1, \ldots, w_{|V_n|}\}$ denote the vertices of V that are adjacent to v in G, and define $E_{vw} = \{(v, w_1), \ldots, (v, w_{|V_n|})\}$ and $V_{vn} = V_n \cup \{v\}$. Assume that there exists a minimally 2-rigid subgraph $G_{vn} = (V_{vn}, E_{vn})$ of G, where $E_{vw} \subset E_{vn}$. Then:

(i) The graph $G_n = (V_n, E_{vn} \setminus E_{vw})$ is a tree.

(ii) For any leaf w of G_n, the graph $G_{c((v,w))}$, which is obtained performing an edge contraction on (v, w) on G, is 2-rigid if G is 2-rigid. $G_{c((v,w))}$ is minimally 2-rigid if G is minimally 2-rigid.

Proposition 2, Proposition 3, and Corollary 1 can be easily applied to find a 2-contractible edge and solve the closing ranks problem, once the stated hypotheses are satisfied, which can be checked using a local test just using information (of links) among the lost agent (removed vertex) and its neighbors.

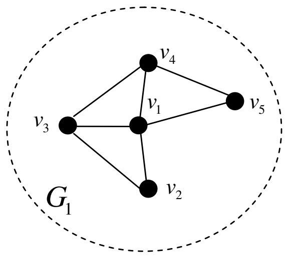

Figure 6. Non-contractible edges: In the minimally 2-rigid graph G_1, (v_1, v_3) and (v_1, v_4) are not 2-contractible.

C. Elimination of Non-Contractible Edges

In this subsection we discuss conditions for an edge of a (minimally) 2-rigid graph not to be 2-contractible. First, we establish a sufficient condition in the following lemma, whose proof is omitted here and can be found in [14].

Lemma 6: Consider a minimally 2-rigid graph $G = (V, E)$. Let $v, w \in V$ and $(v, w) \in E$. If $n_{cn}(v, w) \geq 2$, $n_{cn}(v, w)$ denoting the number of vertices in $V \setminus \{v, w\}$ adjacent to both v and w, then (v, w) is not 2-contractible in G.

Lemma 6 is illustrated in Figure 6: Assume that G_1 is minimally 2-rigid, and consider the closing ranks problem with loss of agent 1 (removal of vertex v_1). In finding an edge to be contracted, applying Lemma 6 with $v = v_1$, (v_1, v_3) and (v_1, v_4) are not 2-contractible, and hence can be eliminated. Then, by Theorem 3, both of the remaining edges (v_1, v_2) and (v_1, v_5) are 2-contractible, i.e., any one of (v_1, v_2) and (v_1, v_5) can be contracted to solve the closing ranks problem.

Lemma 6 provides an easy test requiring only local information. However, the condition in Lemma 6 is only necessary. A more general result involving the concept of *implicit edges* [11] is given in the extended version of this paper, [14].

V. Concluding Remarks

In this paper, we have proposed use of a particular graph operation, the *edge contraction* operation, as a tool for systematic solution of the closing ranks problem, i.e. preservation of rigidity during loss of an agent, for rigid multi-agent formations. We have established, for any vertex v of any 2-rigid graph G, existence of at least two *2-contractible* edges incident on v, contraction of which preserves 2-rigidity. From rigid multi-agent formation perspective, this corresponds to restoration of rigidity of an arbitrary two-dimensional rigid formation, after loss of an agent, by transferring all links to which this agent was incident on to one of its neighbors.

The practical implementation of the above graph operation tool is discussed. Despite establishment of existence of at least two *2-contractible* edges incident on each vertex in any 2-rigid graph in this paper, it is not known yet whether a decentralized algorithm to check if an arbitrary edge is 2-contractible can be designed. It is demonstrated, however, that such an algorithm would need information on vertices and edges that can be at arbitrarily large distance from the edge considered. Later, a set of graph theoretical results are established for several general settings, which can be used in selection of the edge to contract in order to solve the corresponding closing ranks problems.

Discussions and results on extension of the established two-dimensional results to three dimensions can be found in [14], the extended version of this paper. Two other potential future research topics are extension of the results for the agent loss problems involving more than one lost agents, and application of similar tools in agent loss scenarios for robust multi-agent formations with asymmetric control structures [9], [15].

Acknowledgement

NICTA is funded by the Australian Government as represented by the Department of Broadband, Communications and the Digital Economy and the Australian Research Council through the ICT Centre of Excellence Program. J.M. Hendrickx is a postdoctoral fellow of the F.R.S.-FNRS (Fund for Scientific Research, Belgium) and of the Belgian American Education Foundation. His work is also supported by the Concerted Research Action (ARC) “Large Graphs and Networks” of the French Community of Belgium and by the Belgian Programme on Interuniversity Attraction Poles.

References

[1] B. Fidan, B. D. O. Anderson, C. Yu, and J. M. Hendrickx, *Modeling and Control of Complex Systems*, ch. 8. Persistent Autonomous Formations and Cohesive Motion Control. Taylor & Francis, 2007.
[2] A. Das, R. Fierro, V. Kumar, and J. Ostrowski, “A vision-based formation control framework,” *IEEE Trans. on Robotics and Automation*, vol. 18, pp. 813–825, October 2002.
[3] T. Eren, B. Anderson, A. Morse, W. Whiteley, and P. Belhumeur, “Operations on rigid formations of autonoumous agents,” *Communications in Information and Systems*, pp. 223–258, September 2004.
[4] *IET Control Theory and Applications*, vol. 1, March 2007.
[5] W. Ren and R. Beard, “A decentralized scheme for spacecraft formation flying via the virtual structure approach,” *AIAA Journal of Gudiance, Control and Dyanmics*, vol. 27, pp. 73–82, January 2004.
[6] B. Anderson, C. Yu, B. Fidan, and J. Hendrickx, “Rigid graph control architectures for autonomous formations,” *IEEE Control Systems Magazine*, vol. 28, pp. 48–63, December 2008.
[7] S. Martinez and F. Bullo, “Optimal sensor placement and motion coordination for target tracking,” *Automatica*, vol. 42, pp. 661–668, 2006.
[8] A. Bishop, B. Fidan, B. Anderson, K. Dogancay, and P. Pathirana, “Optimality analysis of sensor-target geometries in passive localization,” in *Proc. 3rd Int. Conf. on Intelligent Sensors, Sensor Networks and Information Processing*, pp. 7–18, Dec. 2007.
[9] C. Yu, J. Hendrickx, B. Fidan, B. Anderson, and V. Blondel, “Three and higher dimensional autonomous formations: Rigidity, persistence and structural persistence,” *Automatica*, pp. 387–402, March 2007.
[10] T. Summers, C. Yu, and B. Anderson, “Addressing agent loss in vehicle formations and sensor networks,” *International Journal of Robust and Nonlinear Control*, to appear.
[11] T. Tay and W. Whiteley, “Generating isostatic frameworks,” *Structural Topology*, no. 11, pp. 21–69, 1985.
[12] W. Whiteley, *Matroid Theory*, vol. 197 of *Contemporary Mathematics*, ch. Some Matroids from Discrete Applied Geometry, pp. 171–311. American Mathemtical Society, 1996.
[13] W. Whiteley, *Handbook of Discrete and Computational Geometry*, ch. Rigidity and Scene Analysis, pp. 893–916. CRC Press, 1997.
[14] B. Fidan, J. Hendrickx, and B. Anderson, “Closing ranks in rigid multi-agent formations using edge contraction,” *preprint*, 2009.
[15] J. Hendrickx, B. Fidan, C. Yu, B. Anderson, and V. Blondel, “Primitive operations for the construction and reorganization of persistent formations,” *IEEE Trans. Automatic Control,*, vol. 53, pp. 968–979, May 2008.
[16] L. R. Foulds, *Graph Theory Applications*. Springer-Verlag, 1992.