16th Mediterranean Conference on Control and Automation
Congress Centre, Ajaccio, France
June 25-27, 2008

Decentralized Closing Ranks in Vehicle Formations and Sensor Networks

Tyler H. Summers, Changbin Yu, and Brian D.O. Anderson
The Australian National University and National ICT Australia, Canberra, Australia

Abstract— **In this paper, we present recent results on the *closing ranks problem* in vehicle formations and sensor networks. The *closing ranks problem* is to determine new sensing/communication links in the event of agent failure in order to recover certain properties of the underlying information architecture. We model the information architecture as a graph $G(V,E)$, where V is a set of vertices representing the agents and E is a set of edges representing information flow amongst the agents. We focus on two properties of the graph called *rigidity* and *global rigidity*, which are required for formation shape maintenance and sensor network self-localization, respectively. We show that while previous results permit local repair involving only neighbours of the lost agent, the repair cannot always be implemented using only local information. Utilizing a graph theoretic substitution principle, we present new results that can be applied to make the local repair using only local information. We also describe implementation of the solution and illustrate the ideas through examples.**

I. INTRODUCTION

Autonomous vehicle formations and sensor networks are being progressively deployed to perform a variety of tasks including military reconnaissance and surveillance missions, environmental monitoring, underwater exploration and the like. Interest in these technologies is reflected by the considerable attention from the literature [1], [2], [3], [4], [5], [6], [7], [8]. We consider vehicle formations and sensor networks as collections of agents, each with sensing, communication, and computation capabilities, that work together to accomplish a task.

A primary motivation for using large-scale vehicle formations and sensor networks is robustness to loss of a single agent. Large-scale formations and sensor networks are typically composed of relatively inexpensive agents that may be prone to failure. One could consider addressing agent loss in two different ways: (1) introduce robustness into the information architecture a priori such that agent loss does not destroy desirable properties, or (2) perform a "self-repair" operation in the event of agent loss to recover desirable properties. The former approach is addressed in [9]; in this paper, we shall focus on the latter. For large-scale formations and sensor networks in which centralized computation, communication, and sensing is infeasible, we require the repair to be made in a decentralized way. Here, we use the term "decentralized" to encompass two properties: (1) the formation makes a *local repair* involving only neighbours of a lost agent, and (2) the neighbours perform the repair using only *local information* (independent of formation size).

We model the information architecture with a graph $G(V,E)$, where V is a set of vertices representing agents and E is a set of edges representing information flow amongst the agents. We focus on two properties, called rigidity and global rigidity, which are required for formation shape maintenance and self-localization tasks, respectively, and have received significant attention recently in the literature (see e.g. [1], [2], [3], [7], [8], [10]). The formation shape maintenance task is to maintain all inter-agent distances constant such that the formation moves as a cohesive whole. In this case, the edge set E represents the set of inter-agent distances to be actively held constant via control of individual vehicle motion. If a suitably large and well-chosen set of inter-agent distances is held constant, then all remaining inter-agent distances will be constant as a consequence, thus maintaining formation shape. The self-localization task is to uniquely determine positions for each agent from knowledge of a partial set of inter-agent distances and knowledge of the positions in a global coordinate basis of several agents ("anchors"). In this case, the edge set represents the set of known inter-agent distances. Again, if a suitably large and well-chosen set of inter-agent distances is known, then the remaining inter-agent distances may be uniquely determined. Further, if the positions of three non-collinear agents are also known, then all other agent positions may be uniquely determined. These ideas are further explained in the next section.

The "self-repair" approach is related to the *closing ranks problem*, which is to determine new sensing/communication links in the event of agent loss in order to recover rigidity. In [11], Eren et al present a systematic method to solve the closing ranks problem with local repair. This result draws from a graph theoretic theorem given by Tay and Whiteley in [12] that proves that rigidity can be recovered when a vertex is removed from a rigid graph by adding edges only between neighbours of the lost vertex; in terms of formation

Tyler H. Summers is supported by the Australian-American Fulbright Commission. Changbin Yu and Brian D.O. Anderson are supported by National ICT Australia, which is a national research institute with a charter to build Australias pre-eminent Centre of Excellence for information and communications technology (ICT). NICTA is building capabilities in ICT research, research training and commercialisation in the ICT sector for the generation of national benefit. 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. Email: {tyler.summers,brad.yu,brian.anderson}@anu.edu.au

shape maintenance, this is equivalent to assigning further agent pairs between which the distance should be preserved. Although the method in [11] determines a minimal local repair (minimality being in the sense of adding the least number of links to restore rigidity), it cannot always be implemented using only local information. The reason for this is that the effect of adding a particular link may depend on the information architecture in a non-local way, as described later.

We propose decentralized sub-optimal solutions to solve the closing ranks problem in vehicle formations and sensor networks, providing algorithms to recover both rigidity and global rigidity. The sub-optimality refers to the fact that our solutions potentially add more links than required by a minimal solution (but only up to twice as many). However, it is this trade-off that allows decentralized implementation, highlighting a theme that one must often sacrifice a measure of optimality when faced with certain constraints, such as decentralization. The results rely on a graph theoretic *substitution principle*, originally proposed by Whiteley in [13], that guarantees recovery of certain properties of the underlying information architecture. We describe implementation of the results and illustrate them through algorithms and examples.

The paper is organized as follows: In Section II we present some standard graph theoretic concepts used to model information architectures in vehicle formations and sensor networks. We focus on describing the properties of rigidity and global rigidity. In Section III we formally define the closing ranks problem. We review previous results and then present our new results, which are methods to solve the closing ranks problem in vehicle formations and sensor networks in a decentralized way. In Section IV we discuss implementation and illustrate the new results with algorithms and examples. Finally, Section V provides summarizing and concluding remarks and identifies a few possible problems for future consideration.

II. BACKGROUND

In this section, we summarize the graph theoretic concepts used to model information architectures in vehicle formations and sensor networks. We describe rigidity and global rigidity, which are graph theoretic properties required to perform formation shape maintenance and self-localization, respectively.

A. Rigid Graph Theory

A fundamental task for vehicle formations is maintaining some prescribed geometric shape. Drawing from long-standing traditions in structural engineering and combinatorics (see e.g. [12] and references therein), rigid graph theory has been recently introduced in [1], [2], [3] as a means for describing the information architecture required to maintain formation shape.

A graph $G(V,E)$, where V is a set of vertices and $E \subseteq V \times V$ is a set of edges, provides a useful high-level model of the information architecture. We begin by formally defining a formation $F(G,p)$ as a graph G along with a mapping $p: V \rightarrow \Re^{d|V|}$ that assigns to each vertex a position in d-dimensional Euclidean space. Each agent is abstracted as a vertex in the graph. An edge is present between two vertices in the graph, or equivalently a link is present in the formation, whenever there is active maintenance of the Euclidean distance between the two agents. If a vertex v_j is connected by an edge to v_i, we call v_j a *neighbour* of v_i. The distance is maintained using a control law to govern the motion of one or both agents. Obviously, some appropriate quantities must be sensed for use in controlling distances to neighbours. Specifically, in virtually all formation control algorithms based on distance maintenance, each agent needs to sense relative positions to its neighbours in an arbitrary local coordinate basis. This involves sensing both distances to neighbours and angles to neighbours from some local reference. Alternatively, one could sense distances to neighbours and distances between pairs of neighbours and use the cosine law to obtain the appropriate angles.

Roughly speaking, a rigid formation is one that preserves its shape during a smooth motion, i.e. the distance between every pair of agents remains constant. Consider the function $f: \Re^{d|V|} \rightarrow \Re^{|E|}$ defined by

$$f(p) = [..., ||(p_i - p_j)||^2, ...] \tag{1}$$

where the kth entry of f corresponds to the squared distance between vertices i and j when they are connected by an edge. Now, suppose the formation moves but $f(p)$ stays constant, i.e. the edges in E correspond to links where distance is preserved. Then expanding $f(p)$ about the constant value in a Taylor series and ignoring higher order terms, we obtain

$$J_f(p)\delta p = 0 \tag{2}$$

where δp is an infinitesimal perturbation of the formation, and $J_f(p)$ is the Jacobian of f. This Jacobian is known as the *rigidity matrix* $R(p)$. Equivalently,

$$J_f(p)\dot{p} = 0 \tag{3}$$

for a formation undergoing smooth motion. When the formation is rigid, the only permissible smooth motions are translation or rotation of the whole formation. In d dimensions, this accounts for $(1/2)d(d+1)$ linearly independent vectors. Thus the kernel of $J_f(p)$ has dimension $(1/2)d(d+1)$. This leads us to the following linear algebraic characterization of rigidity:

Theorem 1: A formation $F(G,p)$ is rigid[1] if and only if $rank[R(p)] = d|V| - d(d+1)/2$, which is the maximum rank $R(p)$ can have.

Because the rigidity matrix is a Jacobian of a rational function, it has the same rank for all points but a set of

[1]Actually, the term infinitesimally rigid is sometimes used. See [12] for further discussion of different rigidity concepts.

measure zero (via a nontrivial result of Sard [14]), corresponding to special vertex configurations (e.g. a set of agents are collinear or occupy the same point) that cause the rank deficiency. This leads to the notion of *generic* rigidity. For *generic* configurations (when the special configurations are precluded), information about formation rigidity is contained in the graph, allowing for the drawing of a purely combinatorial consequence of rigidity. The following theorem is presented as the "Necessary Counts Theorem" in [15]:

Theorem 2: If a graph $G(V,E)$ in $\Re^d$ is rigid, then there exists a subset E' of edges such that the induced subgraph $G'(V,E')$ satisfies the following:

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

A graph is called *minimally rigid* if it is rigid and there exists no rigid graph with the same number of vertices and a smaller number of edges. Equivalently, a graph is *minimally rigid* if removing any edge results in loss of rigidity. Intuitively, a minimally rigid graph on a prescribed vertex set must have a minimum number of edges, and the edges must be properly distributed. The set of basis edges corresponding to E' in Theorem 2 are called *independent*. An edge added to a graph is called *dependent* whenever the addition results in a subgraph that violates the second condition in Theorem 2. Equivalently, an edge is called *dependent* when the corresponding row of the rigidity matrix associated with a generic formation connected with the graph is linearly dependent on rows of the matrix present before addition of the edge. When a graph remains rigid after removing any edge, it is called *redundantly rigid*.

In the plane, we have a complete characterization of rigidity due to Laman [16]. In particular, the conditions given in Theorem 2 are both necessary and sufficient for rigidity. Further, there is a set of two basic operations, called Henneberg operations, that allow one to "grow" every minimally rigid graph in the plane from the complete graph on two vertices [17], [12]. Let j and k be two distinct vertices of a minimally rigid graph $G(V,E)$. A *vertex addition* operation involves adding a vertex i and edges ij and ik. Let x, y, and z be three distinct vertices of a minimally rigid graph with edge xy. An *edge splitting* operation involves removing xy and adding a vertex w and edges wx, wy, wz. The operations are illustrated in Figure 1.

A fundamental task for sensor networks is to determine uniquely the position of each agent from knowledge of certain inter-agent distances and the positions of a small number of agents. This task is related to a further concept called *global rigidity*. A graph is called *globally rigid* if two formations having the same inter-agent distances differ at most by translation, rotation, and reflection. In [8], Aspnes et al show that global rigidity is required for unique self-localization in sensor networks (and when the positions of any three non-collinear agents are known, global rigidity is sufficient for localizability of every agent). In [18], Jackson and Jordan prove a conjecture posed by Hendrickson in [19] that provides a complete characterization of global rigidity in the plane. The result is as follows:

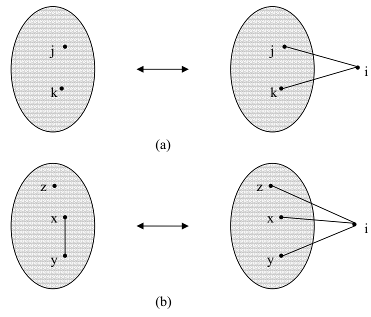

Fig. 1. Representation of (a) vertex addition operation (b) edge splitting operation.

Theorem 3: A graph $G(V,E)$ in the plane is globally rigid if and only if it is 3-connected and redundantly rigid.

A graph is called *minimally globally rigid* if it is globally rigid and there exists no globally rigid graph with the same number of vertices and a smaller number of edges, or equivalently, if removing any edge results in loss of global rigidity. Similarly, the edge splitting operation described above can be used to "grow" all minimally globally rigid graphs from the complete graph on four vertices.

We are interested recovering rigidity or global rigidity so that formation shape maintenance or self-localization can be performed even in the event of agent loss.

III. DECENTRALIZED CLOSING RANKS

In this section, we describe the *closing ranks problem* and show that while previous results permit local repair of the information architecture involving only neighbours of the lost agent, the repair cannot always be implemented using only local information. We then introduce new results that can be implemented in a decentralized way, using only local information. The new results permit decentralized recovery of both rigidity and global rigidity for vehicle formations and sensor networks in two dimensions.

A. The Closing Ranks Problem

The *closing ranks problem* is the problem of determining new sensing/communication links in the event of an agent loss to recover rigidity or global rigidity [11]. Consider a rigid (globally rigid) graph $G(V,E)$ which is the underlying graph of a formation in real d-dimensional space. Let v_i denote a vertex in V, and let E_i denote the set of edges incident to v_i. Now suppose that v_i and E_i are removed from G and denote the resulting graph $G^*(V^*,E^*)$, where $V^* = V \setminus v_i$ and $E^* = E \setminus E_i$. The closing ranks problem is to determine the new edges E_{new} to add to G^* such that the resulting graph

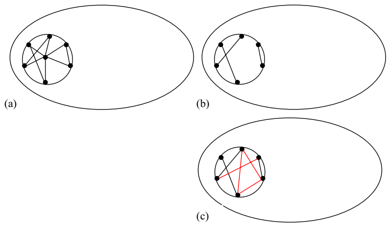

Fig. 2. Illustration of Closing Ranks: (a) original rigid formation (b) a vertex and incident edges are removed (c) the minimum number of new edges necessary for Theorem 2 are added to regain rigidity. It is sufficient to add edges to neighbours of the lost vertex.

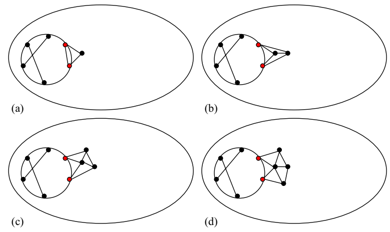

Fig. 3. Edge splitting creates edge dependency between the highlighted vertices. Repeatedly applying edge splitting increases the size of the dependent subgraph.

$G'(V^*, E^* \cup E_{new})$ is rigid (globally rigid). We call a solution a *rigid cover* (*globally rigid cover*). A *minimal cover* is one where $|E_{new}|$ is minimum. A minimal cover for a closing ranks problem is illustrated in Figure 2.

We desire the cover to be decentralized, which encompasses two properties: (1) local repair involving only neighbours of the lost agent, and (2) designing the repair, i.e. determining which agent pairs acquire an edge between them, using only local information. Local repair is provided by the following result from [12]:

Theorem 4: Let G be a minimally rigid graph with a vertex v_i of degree k. Then there exists at least one set of $k-2$ edges among the neighbours of v_i such that a minimally rigid graph is obtained by removing v_i and its incident edges and adding the set of $k-2$ edges.

Thus, it is sufficient to add edges only between neighbours of the lost agent to recover rigidity. However, the difficulty in applying this theorem is to select the $k-2$ edges. If though we drop the requirement for minimality, one straightforward solution to the closing ranks problem is a *complete cover*: simply add every possible edge between neighbours of the lost vertex. Thus, if a vertex of degree k is lost, then $k(k-1)/2$ edges are added ($O(k^2)$). This may be significantly more than the necessary $k-2$ edges for vertices of large degree.

It turns out that the minimal cover cannot always be implemented using only local information. The systematic method presented in [11] to determine a minimal cover may involve decomposing the entire graph, which uses an inherently global perspective. Indeed, when adding only the minimal number of required edges, one must ensure that each added edge is independent. To identify whether a proposed new edge will be independent or not, one must check whether or not there is a minimally rigid subgraph containing the two vertices on which the proposed new edge will be incident. A simple observation shows that generally such a minimally rigid subgraph may be arbitrarily large, and thus looking for it is not a procedure that involves just local operations; given a minimally rigid subgraph on three or more vertices, the edge splitting operation may always be used to increase indefinitely the size of the minimally rigid subgraph. This is illustrated in Figure 3. Further, when checking for edge independence, there is no way to know a priori which subgraph to check, and consequently, one may end up checking the entire graph.

For large formations, it is crucial to be able to implement a closing ranks solution in a decentralized way. There is an inherent conflict between the desire to add a minimum number of new edges and the constraint of using only local information. Thus, we will seek a decentralized cover that is sub-optimal in the sense that we may have to add more than the minimum number of required new edges.

B. Decentralized Rigidity Recovery

We have seen from Theorem 4 that there exists a rigid cover among neighbours of the lost vertex. We now present a substitution principle for rigid graphs, originally introduced in [13].

Theorem 5: Given a rigid graph $G(V,E)$ in d-dimensional space, if for any vertex subset V' the induced subgraph $G'(V',E')$ is replaced with a minimally rigid graph $\bar{G}(V',\bar{E})$ on those vertices (V'), then the modified graph $\tilde{G}(V,\tilde{E})$ where $\tilde{E} = (E \setminus E') \cup \bar{E}$ is also a rigid graph in d-dimensional space.

Here, G corresponds to a graph that has lost a vertex and has been repaired via the method from [11], and V' corresponds to the former neighbours of the lost vertex. Theorem 5 shows that implementing a minimally rigid subgraph, which we refer to as a *minimally rigid patch* ($\bar{G}$ in the theorem), on the former neighbours of a lost vertex will recover rigidity. It is not necessary to first make the repair via the method from [11]; one can immediately implement the minimally rigid patch. The requirement of a minimally rigid patch is only to reduce the number of new edges added; the theorem is equally valid with a rigid patch. Effectively, every minimally rigid patch contains at least one cover which recovers rigidity. Note that the theorem is valid in any dimension. The following proposition provides a way to implement such a patch in 2 dimensions.

Proposition 1: In 2 dimensions, the following "Double Patch" implemented on the neighbours of a lost vertex is a rigid cover for the closing ranks problem.

- **Double Patch:** Choose two vertices among the neighbours of the lost vertex to serve as "coordinators" and

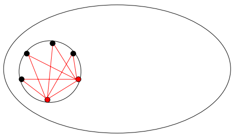

Fig. 4. Illustration of the Double Patch: The two highlighted vertices serve as coordinators to which every other vertex is connected.

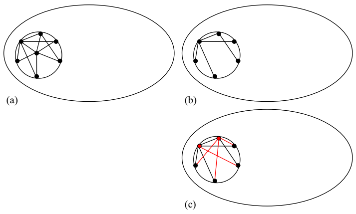

Fig. 5. Utilizing existing cover edges with the Double Patch: (a) the lost vertex has degree 6, (b) edges exist among neighbours of the lost vertex, (c) coordinators for the Double Patch are chosen to utilize these existing edges, and 4 *new* edges are added - a minimal cover.

connect them with an edge. Then add edges between every other neighbour and the coordinators.

The Double Patch is illustrated in Figure 4. This process creates a particular minimally rigid graph on the neighbours of the lost vertex. We are stacking together triangles with a common base edge, which is a rigid structure. Equivalently, we start with a complete graph on the coordinators, and the remaining vertices are added via the vertex addition operation, which is a rigid structure. The choice of the two coordinators is arbitrary and the structure does not depend on the rest of the graph. Thus, the patch can be created using only local information.

When a vertex of degree k is lost, we are adding $2k-3$ edges using the minimally rigid patch. Although we add more edges than with the minimal cover (where $k-2$ edges are added), the number of added edges is still linear in the lost vertex degree, as opposed to quadratic for a complete cover.

By appropriate choice of coordinators, one could utilize existing edges among neighbours of the lost vertex, which we call *existing cover edges*, to minimize the number of *new* added edges. Simple counting arguments can be used to show that there could be up to $k-1$ existing cover edges. In fact, there are certain scenarios in which one adds $2k-3-(k-1)=k-2$ new edges, utilizing the existing cover edges to obtain a minimal cover. This scenario is illustrated in Figure 5.

Remark 1: One might ask whether a "single patch" is sufficient to recover rigidity. That is, could one chose only one vertex among neighbours of the lost vertex as a coordinator? In this case, we add $k-1$ edges - i.e. potentially only one more edge than for a minimal cover. Simulations support the conjecture that there always exists at least one vertex among neighbours of the lost vertex on which a single patch works.

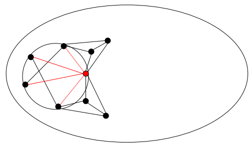

Fig. 6. Single Patch Counterexample. The two dependent edges cause the single patch to fail.

In fact, it has recently asserted to us that in 2 dimensions there are two such vertices. However, we are again faced with the task of choosing the appropriate vertex; once again, local information will not suffice to effect the choice. The counterexample in Figure 6 shows a case where a vertex with no existing cover edges fails using a single patch even though $k-1$ new edges are added. Ultimately, the single patch repair approach fails to be decentralized for the same reason as the minimal cover - one must resort to possibly non-local checking of each new edge for dependency in order to choose the appropriate vertex.

C. Decentralized Global Rigidity Recovery

The ideas discussed above for recovering rigidity can also be applied to recovering global rigidity. The following theorem extends the substitution principle to globally rigid graphs.

Theorem 6: Given a globally rigid graph $G(V,E)$ in d-dimensional space, if any subgraph $G'(V',E')$ is replaced with a globally rigid graph on those vertices (V'), then the modified graph is also a globally rigid graph in d-dimensional space.

Proof.[2] Assume formation $F(G,p)$ is globally rigid, and $G'(p)$ is created by the substitution of a globally rigid formation $F'(p)$ for a sub-formation $F(p)$ in $G(p)$. By Theorem 5, $G'(p)$ is certainly rigid. If $G'(p)$ is not globally rigid, then there is a second realization, i.e. formation whose edge lengths equal those of $G'(p)$, $G'(q)$ which is not congruent. There is a pair of vertices (i,j) for which the corresponding inter-agent distance is different for $G'(p)$ and $G'(q)$. By assumption on F', $F'(p)$ must be congruent to $F'(q)$. So i,j cannot both be in F' (or F). If we now replace $F'(q)$ with $F(q)$ (which will also be congruent to $F(p)$) then we will have $G(p)$ and $G(q)$ with all edge lengths the same, but not congruent, This contradicts the original assumption that $G(p)$ was globally rigid.

Theorem 6 shows that placing a globally rigid subgraph, which we refer to as a *globally rigid patch*, on neighbours of the lost vertex will recover global rigidity. Again, the theorem is valid in any dimension. Proposition 2 provides a way to implement such a patch in 2 dimensions.

[2] Thanks to Professor Walter Whiteley for his help in contributing this proof.

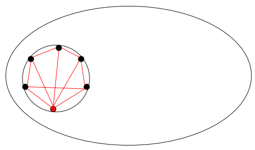

Fig. 7. Illustration of the Wheel Patch: the highlighted vertex serves as the coordinator and connects to every other vertex in the cycle.

Proposition 2: In 2 dimensions, the following "Wheel Patch" implemented on the neighbours of a lost agent is a globally rigid cover for the closing ranks problem.

- **Wheel Patch:** Choose one vertex among the neighbours of the lost agent to serve as "coordinator" and connect it with every other neighbour of the lost agent. Then create a cycle among the neighbours that excluding the coordinator.

The Wheel Patch is illustrated in Figure 7. This process creates a *wheel graph* on the neighbours of the lost agent, which is a particular minimally globally rigid graph. A *wheel graph* on n vertices is a graph that contains a single vertex (the "hub") connected to all vertices of an $(n-1)$-cycle. The choice of hub or coordinator is arbitrary and the structure does not depend on the rest of the graph. Thus, the patch can be created using only local information. Again, the coordinator can be chosen to minimize the number of new edges added.

Remark 2: These results allow decentralized recovery of rigidity or global rigidity while adding only $O(k)$ edges (rather than $O(k^2)$ for a complete cover). One might argue that if minimum vertex degree is large, then connectivity is large, which may lead to redundant rigidity for vertex loss. However, we must distinguish minimum vertex degree, connectivity, and rigidity. In particular, there are graphs with large minimum vertex degree which are neither highly connected nor redundantly rigid.

IV. IMPLEMENTATION

In this section, we describe implementation of our decentralized closing ranks solutions and illustrate with examples. An explicit discussion of underlying assumptions is worthwhile. First, we assume that neighbours of a lost agent recognize the loss immediately. That is, we are not dealing with the separate problem of determining whether agent loss has occurred. Agent loss could occur when an agent has been deployed for another task, or in the event of agent failure. There are in principle many different ways in which an agent could fail, from complete loss of an entire agent (e.g. a UAV crashes) to failure of a single sensor, actuator, or communication link. Of course, recognizing various failure modes is an important consideration for actual implementation, but here we use a "vaporize" agent failure model (that is, regard any failure as complete loss of agent) and assume neighbour agents immediately recognize the loss. Second, we assume that each agent has a unique ID. This is necessary so that the agents can distinguish amongst one another when determining which new links to add.

At each agent store the following local information: a list of links for all neighbours and 2-hop neighbours. This information is local in the sense that it is independent of the size of the formation or sensor network. Since all neighbours of a lost agent will be at most 2-hop neighbours of one another, they each can search the aforementioned list of links to form the subgraph involving all neighbours of the lost agent and existing cover edges. From this subgraph, each neighbour of the lost agent can choose coordinators and determine the new links to add in order to implement one of the patches discussed previously. This can be done in two ways: (1) choose coordinator(s) based on agent ID (e.g. agents with lowest two IDs are coordinators for the double patch), or (2) choose coordinator(s) to minimize number of links added. For the wheel patch, the cycle can also be created using agent ID. Then, the agents establish the appropriate links to recover rigidity or global rigidity. The process is captured in the following algorithms (using agent ID to choose coordinators and cycle order) and illustrated in Figures 8 and 9.

Remark 3: Establishing an link between two agents effectively requires the ability of each agent to *sense* IDs. Existence of a link between two agents means that the agents actively maintain the Euclidean distance constant. As noted before, this involves sensing relative position. Any given agent will have multiple links to maintain and may also have other non-neighbour agents within its sensing range. For each particular link, each agent must be able to distinguish ID through sensing amongst other agents in its sensing range in order to adjust its distance to the appropriate agent.

```
if agent fails then
    for neighbours of lost agent do
        n = get(lost agent neighbours);
        coordinator1 = minID(n);
        coordinator2 = secondMinID(n);
        if I am a coordinator then
            establish links w/ all n;
        else
            establish links w/ coordinators;
        end
    end
end
```

Algorithm 1: Double Patch

V. CONCLUDING REMARKS

In summary, we have presented decentralized solutions for the closing ranks problem in vehicle formations and sensor networks. The results can be used for self-repair in formations and sensor networks in the event of agent loss. We reviewed

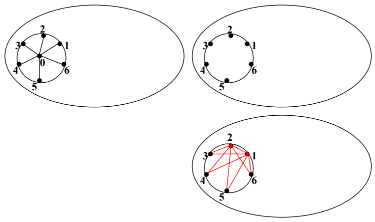

Fig. 8. Double Patch: (a) rigid formation with agent IDs, (b) agent 0 fails, neighbours recognize the loss, (c) coordinators 1 and 2 are chosen via minimum agent ID and links established.

```
if agent fails then
    for neighbours of lost agent do
        n = get(lost agent neighbours);
        coordinator = minID(n);
        cycle = orderID(n\coordinator);
        m = getCycleNeighbours(cycle);
        if I am coordinator then
            establish links w/ all n;
        else
            establish links w/ coordinator & m;
        end
    end
end
```

Algorithm 2: Wheel Patch

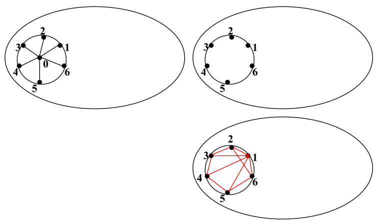

Fig. 9. Wheel Patch: (a) globally rigid formation with agent IDs, (b) agent 0 fails, neighbours recognize the loss, (c) coordinator 1 is chosen via minimum agent ID, cycle formed via agent ID ordering, and links established.

rigid graph theoretic ideas that have been used as high-level models for information architectures. We described an existing solution (the minimal cover) and showed that while repair can be made among neighbours of the lost agent, it cannot be implemented using only local information. By contrast, our solutions are decentralized in two senses: (1) the repair involves only neighbours of the lost agent, and (2) the repair requires only local information, independent of formation size. We described implementation of the results and illustrated with examples.

The ideas in this paper could be extended in several directions. First, as noted previously, another way to deal with losing an agent is to robustify the formation or sensor network a priori by adding certain redundant edges. Suppose, one could obtain this level of robustness; could the closing ranks and patch ideas presented here be used to reestablish the property of redundant rigidity? One could also consider the problem of determining when an agent has failed and how the strategy might change if the failure was just a single sensor or actuator. Finally, one could investigate how the patch ideas presented here relate to how properties such as rigidity and global rigidity could be checked in a decentralized way.

REFERENCES

[1] T. Eren, P. Belhumeur, B. Anderson, and A. Morse, "A framework for maintaining formations based on rigidity," *Proceedings of the 15th IFAC World Congress, Barcelona, Spain*, pp. 2752–2757, 2002.

[2] R. Olfati-Saber and R. Murray, "Graph rigidity and distributed formation stabilization of multi-vehicle systems," *Proceedings of the 41st IEEE Conference on Decision and Control*, vol. 3, pp. 2965–2971, 2002.

[3] J. Baillieul and A. Suri, "Information patterns and Hedging Brockett's theorem in controlling vehicle formations," *Proceedings of the 42nd IEEE Conference on Decision and Control*, vol. 1, pp. 556–563, 2003.

[4] A. Das, R. Fierro, V. Kumar, J. Ostrowski, J. Spletzer, and C. Taylor, "A vision-based formation control framework," *IEEE Transactions on Robotics and Automation*, vol. 18, no. 5, pp. 813–825, 2002.

[5] Z. Lin, M. Broucke, and B. Francis, "Local control strategies for groups of mobile autonomous agents," *IEEE Transactions on Automatic Control*, vol. 49, no. 4, pp. 622–629, 2004.

[6] W. Ren and R. Beard, "Decentralized scheme for spacecraft formation flying via the virtual structure approach," *AIAA Journal of Guidance, Control, and Dynamics*, vol. 27, no. 1, pp. 73–82, 2004.

[7] C. Yu, J. Hendrickx, B. Fidan, B. Anderson, and V. Blondel, "Three and higher dimensional autonomous formations: Rigidity, persistence and structural persistence," *Automatica*, vol. 43, no. 3, pp. 387–402, 2007.

[8] J. Aspnes, T. Eren, D. Goldenberg, A. Morse, W. Whiteley, Y. Yang, B. Anderson, and P. Belhumeur, "A theory of network localization," *IEEE Transactions on Mobile Computing*, vol. 5, no. 12, pp. 1663–1678, 2006.

[9] T. Summers, C. Yu, and B. Anderson, "Robustness to Agent Loss in Vehicle Formations and Sensor Networks," *submitted to the 47th IEEE Conference on Decision and Control*.

[10] A. BDO, Y. C, and F. B, "Information architecture and control design for rigid formations," *2007 Chinese Control Conference*, pp. 2–10, July 26 2007-June 31 2007.

[11] T. Eren, B. Anderson, A. Morse, W. Whiteley, and P. Belhumeur, "Operations on rigid formations of autonomous agents," *Communications in Information and Systems*, vol. 3, pp. 223–258, 2004.

[12] T. Tay and W. Whiteley, "Generating isostatic frameworks," *Structural Topology*, vol. 11, pp. 21–69, 1985.

[13] W. Whiteley, "Infinitesimally Rigid Polyhedra. I. Statics of Frameworks," *Transactions of the American Mathematical Society*, vol. 285, no. 2, pp. 431–465, 1984.

[14] S. A, "The Measure of the Critical Values of Differentiable Maps," *Bulletin of the American Mathematical Society*, vol. 48, pp. 883–890, 1942.

[15] W. Whiteley, "Rigidity and scene analysis," *CRC Press Discrete Mathematics And Its Applications Series*, pp. 893–916, 1997.

[16] G. Laman, "On graphs and rigidity of plane skeletal structures," *Journal of Engineering Mathematics*, vol. 4, no. 4, pp. 331–340, 1970.

[17] L. Henneberg, *Die graphische Statik der starren Systeme*. Teubner, 1911.

[18] B. Jackson and T. Jordán, "Connected rigidity matroids and unique realizations of graphs," *Journal of Combinatorial Theory, Series B*, vol. 94, no. 1, pp. 1–29, 2005.

[19] B. Hendrickson, "Conditions for Unique Graph Realizations," *SIAM Jounal on Computing*, vol. 21, no. 1, pp. 65–84, 1992.