JOURNAL OF AERONAUTICS AND SPACE TECHNOLOGIES
JANUARY 2007 VOLUME 3 NUMBER 1 (21-27)

DIRECTION PROBLEM IN LEADER-FOLLOWER FORMATIONS OF UNMANNED AERIAL VEHICLES AND SATELLITE CLUSTERS

Tolga EREN
Kırıkkale University, Department of Electrical Engineering, 71450 Kırıkkale
tolga.eren@aya.yale.edu

ABSTRACT
This paper is concerned with information structures used in rigid formations of unmanned aerial vehicles (UAV's) and micro-satellite clusters that have leader-follower architecture. The focus of the paper is on sensor/network topologies to secure control of rigidity. Specifically, we study the problem of determining the directions of links of an undirected formation so that the resulting formation is rigid, which is called the "Direction Problem." The algorithm given in the paper establishes a sequential way of determining the directions of links in a leader-follower formation from a given undirected rigid formation. The algorithm is related to our simpler process of finding two spanning trees, as well as the counts in Laman's Theorem. In common with tree finding algorithms, it is greedy, so the order of testing edges does not effect the size of maximal independent sets found, or the distribution of the edges through the UAV's and micro-satellites.

Keywords *Multi-agent systems, UAV formations, satellite clusters, robot formations, rigid formations, graph theory*

1. INTRODUCTION

Recent years have seen significant interest in formations of multiple mobile autonomous agents. (see for example [1-10].) This interest arises from the broad potential for applications, including formation flight, satellite clusters, advanced transportation systems, distributed sensor networks, flocking and schooling, search-and-rescue operations, competitive games, and military reconnaissance and surveillance. In this paper, agents will simply be thought of as autonomous agents including unmanned aerial vehicles and micro-satellites. A *formation* is a group of agents moving in real 2- or 3-space, with some specified links whose distances are maintained. A formation is called *rigid* if the distance between each pair of agents does not change over time under ideal conditions. A formation is called *minimally rigid* if it loses its rigidity when any one of its links is removed from the formation. In other words, a minimally rigid formation has the minimum number of links to maintain rigidity. If a formation is rigid but not minimally rigid, then it is called a *redundantly rigid* formation.

Sensing/communication links are used for maintaining fixed distances between agents. The interconnection structure of sensing/communication links is called *sensor/network topology*. In practice, actual agent groups cannot be expected to move exactly as a rigid formation because of sensing errors, actuation errors, actuation delays, vehicle modelling errors, etc. The ideal benchmark formation against which the performance of an actual agent formation is to be measured is called a *reference formation*.
In reality, agents are entities with physical dimensions. For modeling purposes in this paper, agents are represented by points called *point agents*. Distances between all agent pairs can be held fixed by directly measuring distances between only some agents and keeping them at desired values. A distance constraint or link, is a requirement that a distance between two agents, depicted with d, be maintained through a sensing/communication link and some control strategy. Distance constraints are sometimes referred to as range or separation constraints. With enough distance constraints, the whole formation will be rigid, even without there being a distance constraint between every pair of agents.

Two agents connected by a sensing/communication link are called *neighbors*. There are two types of neighbor relations in rigid formations. In the first type, the neighbor relation is symmetric, i.e., if agent i senses/communicates with agent j and performs action upon the information it receives, so does agent j with agent i. A link with a symmetric neighbor relation is represented graphically by a straight line. In the second type, the neighbor relation is asymmetric, i.e., if agent i senses/communicates with agent j and performs actions upon the information it receives, then agent j does not make use of any information received from agent i although it may sense/ communicate with agent i. For example, rigid formations with a leader-follower architecture have the asymmetric neighbor relation. A link with an asymmetric neighbor relation between a leader and a follower is represented by a directed edge, or arrow, pointing from the follower to the leader, i.e., head is the leader and tail is the follower. The terms “undirected formation” and “directed formation” are used throughout the paper to describe formations with symmetric neighbor relations and formations with leader-follower architecture, respectively.

The work in [2,11,15,16] suggested an approach based on rigidity for maintaining formations of autonomous agents with sensor/network topologies that use distance information between agents, where the neighbor relation is symmetric. Rigidity of undirected formations with distance information is well understood in 2-space, and there are partial results in 3-space [11]. Other researchers focused on using both distance and bearing information to maintain formations that have leader-follower architecture [4]. Formations with directed links were studied in [4,5,12,13,14,17,18].

In this paper, we address the problem of determining the direction of links in a leader-follower formation so that the resulting formation is rigid, which is called the “Direction Problem.” The algorithm given in §5 establishes a sequential way of determining the directions of links from a given undirected rigid formation so that the resulting directed formation is rigid.

2. RIGID FORMATIONS

We start with a brief overview of rigidity. Recall that a formation is rigid if the distance between each pair of agents does not change over time under ideal conditions. It is not necessary to have sensing and communication links between each pair of agents to maintain a rigid formation [11]. Distances between all agent pairs can be held fixed by directly measuring distances between only some agents and keeping them at desired values. We show such an approach for maintaining formations with a limited number of links with distance information both in 2- and 3-dimensional space.

Central to the development of the approach in this section will be rigid frameworks studied in mathematics and engineering for more than a century under different names such as frameworks, linkages, and mechanisms (see for example [19-26]). One way of visualizing rigidity is to imagine a collection of rigid bars connected to one another by idealized ball joints, which is called a bar-joint framework. By an idealized ball joint we mean a connection between a collection of bars which imposes only the restriction that the bars share common endpoints. Now, can the bars and joints be moved in a continuous manner without changing the lengths of any of the bars, where translations and rotations do not count? If so, the framework is non-rigid; if not, it is rigid. The answer depends on factors such as which bars are connected to each other at which ball joints, bar lengths, and the dimensionality of the space in which the framework is placed.

Actual physical bar-joint frameworks can be used in modeling a wide variety of physical structures, including rigid ones such as bridges as well as non-rigid structures such as organic molecules. Appropriate bar-joint frameworks representing such structures could be constructed to test the model for rigidity. However, such a concrete framework is feasible only for 2- and 3-dimensional space, and such concrete models become cumbersome as the number of bars and joints increases. The aim of rigidity theory is to develop methods for predicting rigidity without building a model.

The idea of a point formation is essentially the same as the concept of a “framework” studied in mathematics as well as within the theory of structures in mechanical and civil engineering. For our purposes, a point formation $F_p = (\{p_1, p_2, \ldots, p_n\}, E)$ provides a natural high-level model for a set of n agents moving in real 2- or 3- dimensional space. In this context, the points p_i represent the positions of agents in R^d $\{d = 2$ or $3\}$ and the links in E label those specific agent pairs whose inter-agent distances are to be maintained over time. In practice actual agent positions cannot be expected to move exactly in formation because of sensing errors, vehicle modelling errors, etc. The ideal benchmark formation against which the performance of an actual agent formation is to be measured is called a *reference formation*.

Each point formation F_p uniquely determines a graph $G = (V,E)$ with vertex set $V \{1, 2,\ldots,n\}$ and edge set E, as well as a distance function $\delta : E \rightarrow R$ whose value at $(i, j) \in E$ is the distance between p_i and p_j. Let us note that the distance function of F_p is the same as the distance function of any point formation F_q with the same graph as F_p provided q is congruent to p in the sense that there is a distance preserving map $T : R^d \rightarrow R^d$ such that $T(q_i) = p_i$, $i \in \{1, 2,\ldots,n\}$. In the sequel we will say that two point formations F_p and F_q are congruent if they have the same graph and if q and p are congruent. By a *trajectory* of F_p, we mean a continuously parameterized, one-parameter family of points $\{q(t) : t \geq 0\}$ in R^{nd}, which contains p. A point formation F_p is said to be *rigid* if the distance between every pair of its points remains constant along any trajectory on which the lengths of all of its maintenance links in E are kept fixed. In other words, a point formation F_p is said to be rigid if rigid motion is the only kind of motion it can undergo along any trajectory on which the lengths of all links in E remain constant. Thus, if F_p is rigid, it is possible to “keep formation" by making sure that the lengths of the formation's maintained links do not change as the formation moves. A formation is called *minimally rigid* if it loses its rigidity when any one of its links is removed from the formation.

3. GENERIC RIGIDITY

In this section, we review "generic" rigidity, which is the type of rigidity most useful for our purposes. In practice, actual agent groups cannot be expected to move exactly in rigid formation because of sensing, modeling, and actuation errors. With generic rigidity, the topology will be robust for maintaining formations under small perturbations. A point formation F_p is *generically rigid* if it is rigid for almost all choices of p in R^{dn}. Generic rigidity is a property of only the set of maintenance links, or the underlying graph. It does not even claim that F_p itself is rigid but only that almost all nearby points q give rigid formations F_q. The concept of generic rigidity does not depend on the precise distances between the points of F_p but examines how well the rigidity of formations can be judged by knowing the vertices and their incidences, in other words, by knowing the underlying graph.

For 2-dimensional space, we have a complete combinatorial characterization of generically rigid graphs, which was first proved by Laman in 1970 [24]. In the theorem below, |.| is used to denote the cardinal number of a set, i.e., the number of elements in a set.

Theorem (Laman [24]). *A graph* $G = (V,E)$ *(where* $E \neq \emptyset$ *or* $n > 1$*) is generically rigid in 2-dimensional space if and only if there is a subset* $E' \subseteq E$ *satisfying the following two conditions: (1)* $|E'| = 2|V| - 3$*, (2) For all* $E'' \subseteq L'$, $E'' \neq \emptyset$, $|E''| \leq 2|V(E'')| - 3$*, where* $|V(E'')|$ *is the number of vertices that are end-vertices of the edges in* E''*.*

There is no comparable complete result for 3-dimensional space, though there are useful partial results [12]. Although we lack a characterization in 3-dimensional space, there are sequential techniques to generate rigid classes of graphs both in 2- and 3-dimensional space based on the vertex addition, edge splitting and vertex splitting operations [12].

4. DIRECTED RIGID FORMATIONS

First, we give some definitions from graph theory, which are relevant to all point formations with leader-follower architecture.

A graph in which each edge is replaced by a directed edge is called a *digraph*, also called a directed graph. When there is a danger of confusion, we will call a graph, which is not a digraph, an *undirected graph*. A digraph having no multiple edges or loops (corresponding to a binary adjacency matrix with 0's on the diagonal) is called a simple digraph. A *directed edge*, is written with an ordered pair of end-vertices *(i, j)* representing an edge directed from *i* to *j* and drawn with an arrow from *i* to *j*. Symmetric pairs of directed edges are called *bidirected edges*. In the context of formations, a bidirected edge is equivalent to an undirected edge in the underlying graph of a formation. In formations that have a leader-follower architecture we will use only digraphs with no bidirected edges. The number of edges directed into a given vertex *i* in a digraph *G* is called the *in-degree* of the vertex and is denoted by $d^-_G(i)$. The number of edges directed out from a given vertex *i* in a digraph *G* is called the *out-degree* of the vertex and is denoted by $d^+_G(i)$. The *out-neighborhood* $N^+_G(i)$ of a vertex *i* is $\{j \in V : (i, j) \in E\}$, and the *in-neighborhood* $N^-_G(i)$ of a vertex *i* is $\{j \in V : (j,i) \in E\}$. The union of out-neighborhood and in-neighborhood is the set of *neighbors* of *i*, i.e., the *(open) neighborhood* of *i*, $N_G(i)$. When *i* is also included, it is the *closed neighborhood* of *i*, $N_G[i]$.

In a formation with leader-follower architecture, each link is denoted with an arrow directed from follower to leader. One key type of leader-follower topology is as follows: There is one global leader and one first-follower of the global leader. The global leader does not follow any other agent, and the first follower only follows the global leader, so they are connected with one link pointed from the first-follower to the global leader. The rest of the agents are followers of at least two other agents. Any agent can also be the leader of other agents. We call such an architecture a global leader-first follower architectures. Fig. 1 shows such an example. If global leader-first follower formation is to be rigid, then it is easy to see that ordinary agents (agents other than the global leader and the first follower) must have at least two links. The global leader has 2 degrees of freedom, the first follower has 1 degree of freedom, which makes 3 degrees of freedom in total. This allows these agents to control rigid motions (translation and rotations) of a formation. If any one of the other agents has less than two links, this results in an additional degree of freedom, and the formation need not move rigidly anymore. Recall that the global leader has no outgoing links and the first follower has one link of out-degree 1. Because every other agent is at least of out-degree 2, we have at least $2(n - 2) + 1 = 2n - 3$ links in total.

With a generically minimally rigid graph, with global leader-first follower architecture, all other vertices will have degree exactly 2. We give an explicit algorithm for this in detail in §5.

A digraph G = (*V, E*) is 2-*directed* if for all $i \in V$, $|N^+_G(i)| \leq 2$. In a 2-directed digraph G = (*V, E*) that has global leader-first follower architecture with count $|E| = 2|V|-|3|$, all vertices except the ones that correspond to the global leader and first follower will have out-degree of exactly two. Even such a digraph may not have an underlying rigid graph. A formation is *directed rigid* if it is minimally rigid and 2-directed [18].

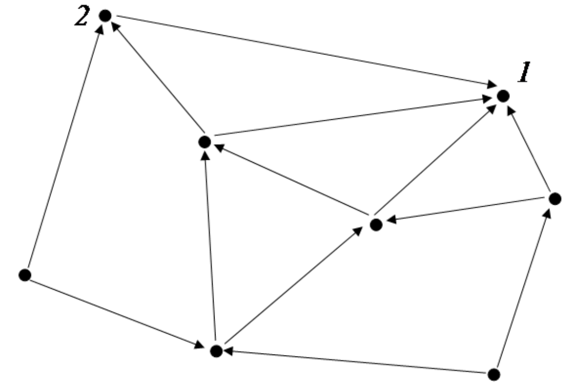


Figure 1. Global leader-first follower architecture: All vertices are of out-degree 2, except that there is one vertex of out-degree 0 (labeled with 1), and another vertex of out-degree 1 (labeled with 2), and these two vertices are neighbors.

5. RESULTS: CREATING A DIRECTED RIGID FORMATION FROM AN UNDIRECTED RIGID FORMATION

Directed rigidity of a formation depends not only on the underlying undirected formation but also on the directions of links between agents. In particular, the directed formation must be 2-directed. Given a generically minimally rigid undirected formation, how do we find the directions of links to create a stably rigid directed formation? Below we present one way of doing this. We start with giving preliminary definitions.

A graph is *connected*, if there is a path from any vertex to any other vertex in the graph. A *tree* is a graph in which any two vertices are connected by exactly one path. A *spanning tree* of a connected, undirected graph is a tree which includes every vertex of that graph. There is a standard way of partitioning the edges in a generically minimally rigid graph with the following properties:

1) there are three trees;
2) there are exactly two trees at each vertex;
3) no two non-empty subtrees span the same set of vertices.

These properties define a 3Tree2 partition of the edges [21], [25], [26]. For a generically minimally rigid graph $G = (V,E)$, it is also known that, for each $(i,j) \in E$, the multigraph obtained by doubling the edge (i,j) is the union of two spanning trees [21], [27].

Now we give a sequential algorithm to find the direction of links to create a stably minimally rigid directed formation from a minimally rigid undirected formation: (Let us assume that *i* represents the global leader, *j* represents the first follower connected to *i* by the edge *(j,i)*.)

Algorithm: *2-Direction of a Minimally Rigid Formation.*
1) Double the edge (j,i) - The entire graph can now be partitioned into two spanning trees.
2) Remove (j,i) from one of the two trees - We now have 3-trees, one spanning, and one each containing the original two vertices.
3) Orient the spanning tree down to the selected leader.
4) Orient each of the other two trees down to the global leader or the first follower, whichever is in this revised tree.

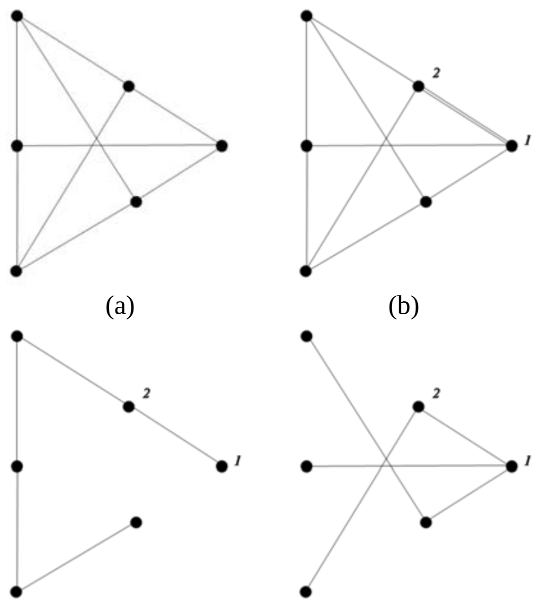
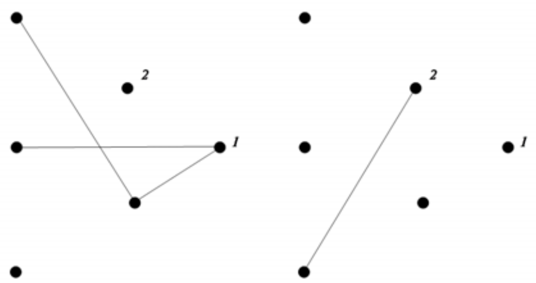


Figure 2. Obtaining two spanning trees: A minimally rigid point formation is shown in (a). The graph with the double edge (2,1) is shown in (b). The global leader is labeled with 1 and the first follower is labeled with 2. The graph in (b) can be partitioned into two spanning trees as shown in (c) and (d).

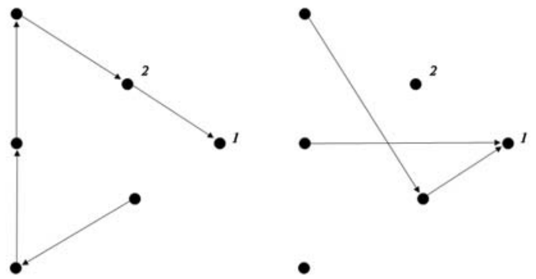
This algorithm gives a directed rigid directed formation with out-degree 2 at each point except the first-follower of out-degree 1 and the global leader of out-degree 0. We give the following example to illustrate this algorithm:

Example: Consider the generically minimally rigid point formation shown in Figure 2(a). Assume that the global leader is labeled with 1 and the first follower is labeled with 2. The graph with the double edge (2,1) is shown in Fig. 2(b). This graph can be partitioned into two spanning tress as shown in Figs. 2(c) and 2(d). When we remove (2,1) from one of the two trees, in this case from Fig. 2(d), we now have three trees: one spanning as shown in Fig. 2(c), and one each containing the original two vertices as shown in Figures 3(a) and 3(b). Fig. 4(a) shows the oriented spanning tree down to the global leader. Figs. 4(b) and 4(c) show the oriented two trees down to the global leader or the first follower. Finally, if we put together the edge topologies in Figs. 4(a), 4(b), and 4(c), we obtain the directed point formation shown in Fig. 4(d).

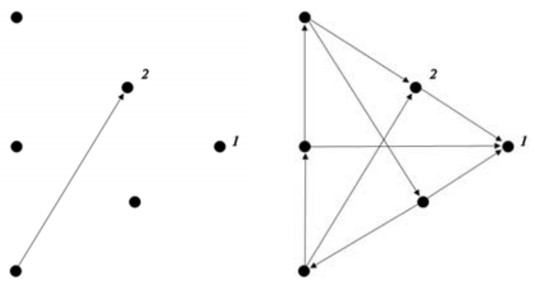


(a) (b)

Figure 3. Obtaining three trees: When we remove (2,1) from one of the two spanning trees in Figs. 2(c) and 2(d), in this case from Fig. 2(d), we now have three trees: one spanning as shown in Fig. 2(c), and one each containing the original two vertices as shown in (a) and (b) in this figure.



(a) (b)



(c) (d)

Figure 4. Orienting the spanning tree: The oriented spanning tree down to the global leader is shown in (a). The oriented two trees down to the global leader or the first follower are shown in (b) and (c). If we put together the edge topologies in (a), (b) and (c), we obtain the directed point formation shown in (d).

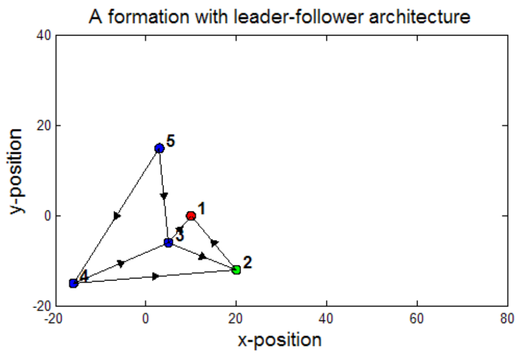
We note that this algorithm permits an arbitrary choice of the first edge in the graph. There is also a way to deduce this decomposition directly from the assumption that the rigidity matrix has independent rows and full rank [26]. Given a rigid graph G, there is a refined fast (worst case $O(|V||E|)$) implemented algorithm (the pebble game) which:

▪ selects a minimally rigid sub-graph in $O(|V||E|)$ time and gives an orientation towards a selected leader-follower edge with out-degree 2 on all other vertices for any minimally rigid graph;
▪ can switch from one such choice of leader-follower edge in a minimally rigid graph to an orientation towards another leader-follower edge in linear time, by cascading pebbles;
▪ can detect whether there is an acyclic 2-directed orientation towards a given leader-follower edge.

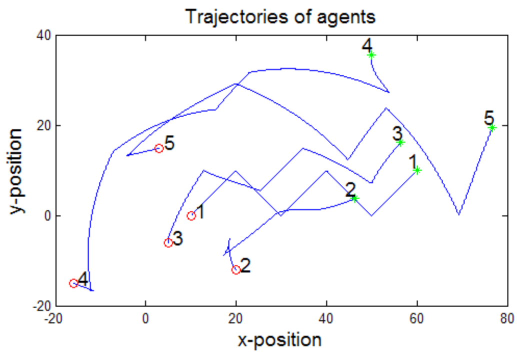
The algorithm is related to our simpler process of finding two spanning trees, as well as the counts in Laman’s Theorem. In common with tree finding algorithms, it is greedy, so the order of testing edges etc. does not effect the size of maximal independent sets found, or the distribution of the edges through the agents. The normal implementation can give, as an immediate output, the desired 2-directed graph. Given some set of vertices and independent edges, the algorithm can also select additional edges to extend this to an (oriented) minimally rigid graph, in order $|V|^2$ time.

There is a third way to generate the digraph. Given a minimally rigid graph, there is a Henneberg sequence [11] starting with the selected global leader-first follower edge. Applied as directed vertex addition and edge splitting, this generates a stably rigid directed formation. Combined with arbitrary cascades of pebbles, these give all possible directed rigid formations. There are order $O(|V|^2)$ algorithms for directly extracting either a 3Tree2 covering or the Henneberg sequence from the minimally rigid graph. However, some recent implementations for these actually use the pebble game as their core engine.

Simulation on a directed formation generated by the algorithm presented above is shown in Fig. 5 (a) and (b). In this simulation, the global leader is denoted with 1, the first follower is denoted with 2, ordinary agents are denoted with 3, 4, 5 in Fig. 5(a). As the global leader moves on a zigzag trajectory as shown in Fig. 5(b), other agents move in such a way that the entire formation maintains its rigidity, i.e., inter-agent distances are preserved.



(a)



(b)

Figure 5. Simulation of a directed formation: The links of this formation is created using the algorithm given in §5. We test the rigidity of the formation in a simulation by moving the global leader on an arbitrary trajectory. (a) The global leader is denoted with 1, the first follower is denoted with 2, and ordinary agents are denoted with 3, 4, 5. (b) The global leader moves along a zigzag trajectory. The remaining agents move in such a way that inter-agent distances are preserved.

6. CONCLUSION

The algorithm given in §5 establishes a sequential way of determining the directions of links from a given undirected rigid formation so that the resulting formation is directed rigid. We anticipate that the pattern of analysis given in this paper will be useful in the analysis of formation rigidity and stability problems and will be a useful tool to create directed rigid formations.

A sequel will give analogs with rigid formations in 3-space, with other types of information structures such as the following: formations with directions, bearings (a.k.a. angle of arrival) in both 2- and 3-space; formations with mixed directions-distances or bearings-distances in 2- and 3-space; formations with mixed directed/undirected links.

7. REFERENCES

[1] Belta, C., and Kumar, V., "Abstraction and control for groups of robots," *IEEE Transactions on Robotics*, vol. 20, no. 5, pp. 865–875, 2004.

[2] Olfati-Saber, R., and Murray, R., "Graph rigidity and distributed formation stabilization of multi-vehicle systems," in *Proceedings of the 41st IEEE Conference on Decision and Control*, Las Vegas, NV, December 2002.

[3] Tabuada, P., Pappas, G., and Lima, P., "Feasible formations of multi-agent systems," *IEEE Transactions on Robotics*, vol. 21, no. 3, pp. 387–392, 2005.

[4] Tanner, H., Pappas, G. J., and Kumar, V., "Leader-to-formation stability," *IEEE Transactions on Robotics and Automation*, vol. 20, no. 3, pp. 443–455, June 2004.

[5] Baillieul, J., and Suri, A., "Information patterns and hedging Brockett's theorem in controlling vehicle formations," in *Proceedings of the 42nd IEEE Conference on Decision and Control*, Maui, Hawaii, December 2003.

[6] Zhang, F., Grocholsky, B., and Kumar, V., "Formations for localization of robot networks," in *Proceedings of the 2004 IEEE International Conference on Robotics and Automation*, April 2004.

[7] Ren, W., and Beard, R. W., "A 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, January 2004.

[8] Lin, Z., Francis, B., and Maggiore, M., "Necessary and sufficient graphical conditions for formation control of unicycles," *IEEE Trans. Automatic Control*, vol. 50, no. 1, pp. 121–127, 2005.

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

[10] Gazi, V., and Passino, K., "Stability analysis of swarms," *IEEE Trans. On Automatic Control*, vol. 48, no. 4, pp. 692–697, 2003.

[11] Eren, T., Anderson, B.D.O., Morse, A.S., Whiteley, W., and Belhumeur, P.N., "Operations on rigid formations of autonomous agents," *Communications in Information and Systems*, vol. 3, no. 4, 2004.

[12] Eren, T., Whiteley, W., Anderson, B. D. O., Morse, A. S., and Belhumeur, P. N., "Information structures to secure control of rigid formations with leader-follower architecture," in *Proceedings of the American Control Conference*, Portland, Oregon, June 2005, pp. 2966–2971.

[13] Yu, C., Fidan, B., and Anderson, B. D. O., “Persistence acquisition and maintenance for autonomous formations,” in Proceedings of Workshop on Unmanned Aerial Vehicles, 2nd International Conference on Intelligent Sensors, Melbourne, Australia, December 2005, pp. 379–384.

[14] Marshall, J., Broucke, M., and Francis, B., “Formations of vehicles in cyclic pursuit,” *IEEE Trans. Automatic Control*, vol. 49, no. 11, pp. 1963–1974, 2004.

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

[16] Eren, T., Belhumeur, P.N., and Morse, A.S., “Closing ranks in vehicle formations based on rigidity,” in *Proceedings of the 41st IEEE Conference on Decision and Control*, December 2002, pp. 2959–2964.

[17] Eren, T., Whiteley, W., and Belhumeur, P.N., “Rigid formations with leader-follower architecture,” Columbia University, Tech. Rep. CUCS-010-05, March 2005. [Online]. Available: http://www1.cs.columbia.edu/~library/

[18] Eren, T., Whiteley, W., and Belhumeur, P.N.,, “Rigid formations with leader-follower architecture,” Columbia University, Tech. Rep. CUCS-008-06, December 2005. [Online]. Available: http://www1.cs.columbia.edu/~library/

[19] Whiteley, W., “Rigidity and scene analysis,” in *Handbook of Discrete and Computational Geometry*, J. Goodman and J. O’Rourke, Eds. CRC Press, 1997, pp. 893–916.

[20] Roth, B., “Rigid and flexible frameworks,” *American Mathematical Monthly*, vol. 88, pp. 6–21, 1981.

[21] Whiteley, W., “Matroids from discrete geometry,” in *Matroid Theory*, J. E. Bonin, J. G. Oxley, and B. Servatius, Eds. American Mathematical Society, Contemporary Mathematics, 1996, vol. 197, pp. 171–313.

[22] Connelly, R., “On generic global rigidity,” June 2003, preprint, Cornell University, Ithaca, New York, USA.

[23] Servatius, B., and Servatius, H., “Generic and abstract rigidity,” in *Rigidity Theory and Applications*, M. Thorpe and P. Duxbury, Eds. Kluwer Academic/Plenum Publishers, 1999, pp. 1–19.

[24] Laman, G., “On graphs and rigidity of plane skeletal structures,” *Journal of Engineering Mathematics*, vol. 4, pp. 331–340, 2002.

[25] Hendrickson, B., “An algorithm for two-dimensional rigidity percolation: The pebble game,” *Journal of Computational Physics*, vol. 137, pp. 346–365, 1997.

[26] Jacobs, D., and Thorpe, M., “Generic rigidity percolation: The pebble game,” *Physical Review Letters*, vol. 75, no. 22, pp. 4051–4054, 1995.

VITAE

Tolga EREN

Tolga Eren received the B.S. degree in electrical engineering from Bilkent University, Ankara, Turkey, the M.S.E.E. degree in electrical engineering from the University of Massachusetts, the M.S., the M.Phil. and the Ph.D. degrees in electrical engineering from Yale University, New Haven, Connecticut, in 1994, 1998, 1999, 2002, and 2003, respectively. From October 2003 to July 2005, he was a postdoctoral research scientist in the Computer Science Department at Columbia University in the City of New York. Since September 2005, he has been in the department of Electrical Engineering at Kırıkkale University, Turkey. His research interests are multi-agent (multi-robot, multi-vehicle) systems, sensor networks, and computer vision.