

Rigidity Maintenance Control for Multi-Robot Systems

Daniel Zelazo*, Antonio Franchi†, Frank Allgöwer*, Heinrich H. Bühlhoff‡, and Paolo Robuffo Giordano†

*Institute for Systems Theory and Automatic Control, University of Stuttgart, Germany.

Email: {daniel.zelazo, allgower}@ist.uni-stuttgart.de

†Max Planck Institute for Biological Cybernetics, Tübingen, Germany.

Email: {antonio.franchi, prg}@tuebingen.mpg.de

‡Department of Brain and Cognitive Engineering, Korea University. Email: hhb@tuebingen.mpg.de

Abstract—Rigidity of formations in multi-robot systems is important for formation control, localization, and sensor fusion. This work proposes a rigidity maintenance gradient controller for a multi-agent robot team. To develop such a controller, we first provide an alternative characterization of the rigidity matrix and use that to introduce the novel concept of the *rigidity eigenvalue*. We provide a necessary and sufficient condition relating the positivity of the rigidity eigenvalue to the rigidity of the formation. The rigidity maintenance controller is based on the gradient of the rigidity eigenvalue with respect to each robot position. This gradient has a naturally distributed structure, and is thus amenable to a distributed implementation. Additional requirements such as obstacle and inter-agent collision avoidance, as well as typical constraints such as limited sensing/communication ranges and line-of-sight occlusions, are also explicitly considered. Finally, we present a simulation with a group of seven quadrotor UAVs to demonstrate and validate the theoretical results.

I. INTRODUCTION

Multi-robot systems provide advantages over their monolithic counterparts for many reasons. Perhaps of greatest interest is their resilience against system failures and their ability to adapt to highly dynamic and unknown environments. Applications that rely on multi-agent solutions vary from interferometry in deep space, distributed sensing and data collection, surveillance, construction and transportation, and search and rescue operations [1, 2, 7, 22, 23, 24, 25]. Despite the many advantages of multi-agent systems, there remain several open problems towards the complete autonomy of such systems, among which control of the agents collective motion is of paramount importance. In this context, the alignment of theoretical and analytical tools with the constraints of real-world systems remains challenging. Limited sensing and communication ranges, inter-agent and obstacle collision avoidance, and connectivity maintenance all represent fundamental requirements for the successful implementation of any collective motion control strategy. In this direction, there has been a focused research effort on the development of distributed algorithms for collective motion control while guaranteeing maintenance of such mission objectives.

An important objective for collective motion control of multiple agents that has not received much attention in the robotics community is that of *rigidity* and *rigid formations*. The study of rigidity has a long history with contributions from both pure mathematics and engineering disciplines [9, 18, 20, 21, 29, 31]. One benefit from rigidity theory in the context of motion control of multiple robots is that it allows for

formation control using only *relative distance measurements*, as opposed to *relative position measurements* from a global or relative inertial frame [2, 3, 5, 20, 26, 30]. For example, in [20] it was shown that formation stabilization using only distance measurements can be achieved only if rigidity of the formation is maintained. In fact, rigidity represents a necessary condition for estimating relative positions using only relative distance measurements [4, 8]. Therefore, in the context of formation control and estimation, the notion of rigidity plays the same role as that of connectivity when distance measurements are the only available information (rather than relative or absolute position measurements).

In a broader context, rigidity turns out to be an important architectural property of many multi-agent systems when a common inertial reference frame is unavailable. Applications that rely on sensor fusion for localization, exploration, mapping and cooperative tracking of a target, all can benefit from notions in rigidity theory [4, 29, 32]. While equipping each robot with a GPS-like sensor may eliminate many difficulties, these solutions would not work in harsher environments, such as under-water, indoors, or any GPS-denied environments. Furthermore, coordination via relative sensing inherently provides greater accuracy and reliability in addition to more resilience and robustness against failures.

A goal of this paper, therefore, is to present a distributed gradient-based control strategy for a group of mobile robots for maintaining *formation rigidity* at all times while navigating in cluttered environments. In particular, we do not aim for a solution constraining the robots to keep a *given fixed rigid formation*, e.g., by choosing beforehand a particular topology for the interaction graph. Rather, we are interested in devising a flexible strategy that can allow, for instance, creation or disconnection of interaction links among robots, while still ensuring rigidity of the entire formation. In this way, the robot motion will not be overly constrained: this is a desirable feature when, for example, establishment/maintenance of an interaction link among two agents conflicts with typical sensing/communication issues such as limited ranges or line-of-sight occlusions.

In general, rigidity as a property of a given formation (i.e., of the robot spatial arrangement) has been studied from either a purely combinatorial perspective [21], or by providing an algebraic characterization via the state-dependent *rigidity matrix* [31]. One contribution of this work, therefore, is the

development of an alternative representation of the rigidity matrix that separates the underlying graph structure from the position of each agent. This is instrumental to create a companion matrix, the *symmetric rigidity matrix*, which will be shown to resemble a weighted Laplacian matrix. Using the symmetric rigidity matrix, we are then able to determine whether a formation is rigid by examining a particular eigenvalue of this matrix, named the *rigidity eigenvalue*. This quantity is in fact, in the planar case, identical to the *worst-case rigidity index*, introduced in [34]. However, contrary to the results presented in [19, 34], the structure of the symmetric rigidity matrix allows for using the rigidity eigenvalue and its associated eigenvector in a distributed control strategy for ensuring rigidity under time-varying topologies and additional sensing constraints. In particular, we will show how to embed into a unified gradient-based *rigidity maintenance action* the fulfillment of a number of additional constraints, such as inter-robot and obstacle avoidance, limited communication and sensing ranges, and line-of-sight occlusions. Our approach, therefore, can be seen as the suitable recasting into the rigidity framework of those works related to decentralized connectivity maintenance based on the second smallest eigenvalue of the graph Laplacian matrix, see, e.g., [27, 33].

The organization of this paper is as follows. Section I-A provides a brief overview of some notation and fundamental theoretical properties of graphs. In Section II, the theory of rigidity is introduced. This section also introduces the first main results of this paper, being the characterization of the symmetric rigidity matrix and of the rigidity eigenvalue. Then, Section III shows how to exploit these results in order to derive the sought gradient-based rigidity maintenance controller. The validity of this controller is further illustrated in Section IV by means of a realistic simulation involving 7 quadrotor UAVs navigating in a cluttered environment. During the simulation, the motion of two of these UAVs is also partially controlled by two human operators through two joysticks in order to stress the action of the controller in maintaining formation rigidity. Finally, Section V concludes the paper and provides some final remarks.

A. Preliminaries and Notations

The notation employed is standard. Matrices are denoted by capital letters (e.g., A), and vectors by lower case letters (e.g., x). The rank of a matrix A is denoted $\mathbf{rk}[A]$. Diagonal matrices will be written as $D = \mathbf{diag}\{d_1, \dots, d_n\}$; this notation will also be employed for block-diagonal matrices. A matrix and/or a vector that consists of all zero entries will be denoted by $\mathbf{0}$; whereas, ‘0’ will simply denote the scalar zero. Similarly, the vector $\mathbf{1}$ denotes the vector of all ones. The $n \times n$ identity matrix is denoted as I_n . The set of real numbers will be denoted as \mathbb{R} , and $\|\cdot\|$ denotes the standard Euclidean 2-norm for vectors. The Kronecker product of two matrices A and B is written as $A \otimes B$ [15].

Graphs and the matrices associated with them will be widely used in this work. The reader is referred to [12] for a detailed treatment of the subject. An undirected (simple) weighted

graph \mathcal{G} is specified by a vertex set \mathcal{V} , an edge set \mathcal{E} whose elements characterize the incidence relation between distinct pairs of \mathcal{V} , and diagonal $|\mathcal{E}| \times |\mathcal{E}|$ weight-matrix A , with $A_{ii} \geq 0$ the weight on edge $e_i \in \mathcal{E}$. Two vertices i and j are called *adjacent* (or neighbors) when $\{i, j\} \in \mathcal{E}$; we denote this by writing $i \sim j$. An *orientation* of an undirected graph \mathcal{G} is the assignment of directions to its edges, i.e., an edge e_k is an ordered pair (i, j) such that i and j are, respectively, the initial and the terminal nodes of e_k .

The incidence matrix $E(\mathcal{G}) \in \mathbb{R}^{|\mathcal{V}| \times |\mathcal{E}|}$ is a $\{0, \pm 1\}$ -matrix with rows and columns indexed by the vertices and edges of \mathcal{G} such that $[E(\mathcal{G})]_{ik}$ has the value ‘+1’ if node i is the initial node of edge e_k , ‘-1’ if it is the terminal node, and ‘0’ otherwise. The degree of vertex i , d_i , is the cardinality of the set of vertices adjacent to it. The degree matrix, $\Delta(\mathcal{G})$, and the adjacency matrix, $A(\mathcal{G})$, are defined in the usual way [12]. The (graph) Laplacian of \mathcal{G} , $L(\mathcal{G}) = E(\mathcal{G})E(\mathcal{G})^T = \Delta(\mathcal{G}) - A(\mathcal{G})$, is a rank deficient positive semi-definite matrix. One of the most important results from algebraic graph theory in the context of collective motion control states that a graph is connected if and only if the second smallest eigenvalue of the Laplacian is positive [12].

II. RIGIDITY AND THE RIGIDITY EIGENVALUE

In this Section we review the fundamental concepts of graph rigidity. The main construct from rigidity theory that we focus on is the rigidity matrix. A main contribution of this work is the presentation of a companion matrix to the rigidity matrix, and the characterization of what we term the *rigidity eigenvalue*. For a more detailed treatment of the theory of graph rigidity, the reader is referred to [13, 17].

A. Graph Rigidity and the Rigidity Matrix

We consider graph rigidity from what is known as a *d-dimensional bar-and-joint framework*. A framework is the pair (\mathcal{G}, p) , where $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ is a graph, and $p : \mathcal{V} \mapsto \mathbb{R}^d$ maps each vertex to a point in \mathbb{R}^d . For simplicity of exposition, we only consider the planar case, $d = 2$, and note that all the following results can be extended to $d = 3$ in a straightforward way. Therefore, for node $u \in \mathcal{V}$, $p(u) = [p_u^x \ p_u^y]^T$ is the position vector in \mathbb{R}^2 for the mapped node. We denote by $p^{(x,y)} = [p(v_1) \cdots p(v_{|\mathcal{V}|})]^T$ as the aggregate position vector of all agents. In fact, although the following derivations only consider the planar case, the simulations presented in Section IV are run by taking into account full 3-dimensional agents and their associated 3-dimensional controller. We now provide some basic definitions.

Definition 2.1: Frameworks (\mathcal{G}, p_0) and (\mathcal{G}, p_1) are *equivalent* if $\|p_0(u) - p_0(v)\| = \|p_1(u) - p_1(v)\|$ for all $\{u, v\} \in \mathcal{E}$, and are *congruent* if $\|p_0(u) - p_0(v)\| = \|p_1(u) - p_1(v)\|$ for all $u, v \in \mathcal{V}$.

Definition 2.2: (\mathcal{G}, p_0) is *globally rigid* if every framework which is equivalent to (\mathcal{G}, p_0) is congruent to (\mathcal{G}, p_0) .

Definition 2.3: (\mathcal{G}, p_0) is *rigid* if there exists an $\epsilon > 0$ such that every framework (\mathcal{G}, p_1) which is equivalent to (\mathcal{G}, p_0)

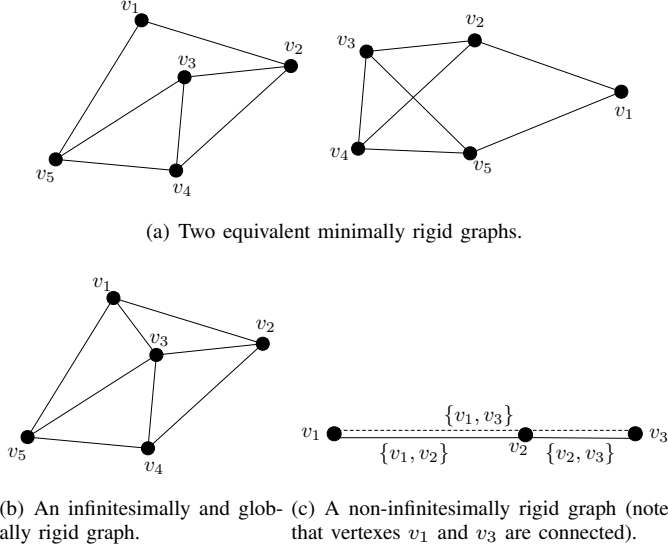


Fig. 1. Examples of rigid and infinitesimally rigid frameworks.

and satisfies $\|p_0(v) - p_1(v)\| < \epsilon$ for all $v \in \mathcal{V}$, is congruent to (\mathcal{G}, p_0) .

Definition 2.4: A *minimally rigid graph* is a rigid graph such that the removal of any edge results in a non-rigid graph.

Figure 1 shows three graphs illustrating the above definitions. The graphs in Figure 1(a) are both minimally rigid and are equivalent to each other, but are not congruent, and therefore not globally rigid. By adding an additional edge, as in Figure 1(b), the graph becomes globally rigid. The key feature of global rigidity, therefore, is that the distances between *all* node pairs are maintained for different framework realizations, and not just those defined by the edge set.

Using the above definitions, a framework (\mathcal{G}, p) can be described as the set

$$\{p(u) \in \mathbb{R}^2 \mid \|p(u) - p(v)\|^2 = \ell_{uv}^2, \forall \{u, v\} \in \mathcal{E}\}, \quad (1)$$

where the Euclidean distances between nodes, ℓ_{uv} , are specified. At a particular point in the configuration space, one can assign velocity vectors $\xi(u) \in \mathbb{R}^2$ to each vertex $u \in \mathcal{V}$ such that

$$(\xi(u) - \xi(v))^T (p(u) - p(v)) = 0, \forall \{u, v\} \in \mathcal{E}. \quad (2)$$

Note that this relation can be obtained by differentiating the length constraint described in (1). These motions are referred to as *infinitesimal motions* of the mapped vertices $p(u)$. If the mapping p is further parameterized by a positive scalar representing time, then we can consider the infinitesimal motions at each time, and define

$$\dot{p}(u, t) = \xi(u), \quad (3)$$

where $\xi(u)$ will be treated as the *agent velocity input* for control purposes (see Section III).

Definition 2.5: A framework is called *infinitesimally rigid* if every possible motion that satisfies (2) is trivial (i.e., consists of only rotations and translations of the entire framework).

An example of an infinitesimally rigid graph is shown in Figure 1(b). Furthermore, note that infinitesimal rigidity implies rigidity, but the converse is not true [31], see Figure 1(c) for a rigid graph that is not infinitesimally rigid.

Infinitesimal rigidity can also be determined by examining the rank of the *rigidity matrix*, $R(p) \in \mathbb{R}^{|\mathcal{E}| \times 2|\mathcal{V}|}$ [31]. In fact, the rigidity matrix can be constructed from the system of equations (2). This can be thought as a way to generate all possible infinitesimal rigid motions of a formation. The quantity $(p(u) - p(v))$, therefore, represents the coefficients of that system of equations, and each row of the matrix corresponds to an edge $e = \{u, v\}$; for example, the row corresponding to edge e has the form

$$\begin{bmatrix} \mathbf{0} & \underbrace{(p(u) - p(v))^T}_{\text{vertex } u} & \mathbf{0} & \underbrace{(p(v) - p(u))^T}_{\text{vertex } v} & \mathbf{0} \end{bmatrix}.$$

The definition of infinitesimal rigidity can then be restated in the following form:

Lemma 2.1 ([31]): A framework (\mathcal{G}, p) is infinitesimally rigid if and only if $\text{rk}[R(p)] = 2|\mathcal{V}| - 3$.

Note that, as expected from Definition 2.5, for an infinitesimally rigid graph the three-dimensional kernel of $R(p)$ only allows for three independent *feasible* framework motions, that is, the above-mentioned collective roto-translation on the plane. Note also that, despite its name, the rigidity matrix is actually characterizing *infinitesimal rigidity* rather than *rigidity* of a framework.

A first contribution of this work is to provide an alternative representation of the rigidity matrix that separates the underlying graph from the relative positions of each agent. In this direction, we first define the notion of a *directed local graph* at node v_i .

Definition 2.6: Consider a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ and its associated incidence matrix with arbitrary orientation $E(\mathcal{G})$. The *directed local graph* at node v_j is the sub-graph $\mathcal{G}_j = (\mathcal{V}, \mathcal{E}_j)$ induced by node v_j such that

$$\mathcal{E}_j = \{(v_j, v_i) \mid e_k = \{v_i, v_j\} \in \mathcal{E}\}.$$

The *local incidence matrix* at node v_j is the matrix

$$E_l(\mathcal{G}_j) = E(\mathcal{G}) \mathbf{diag}\{s_1, \dots, s_{|\mathcal{E}|}\} \in \mathbb{R}^{|\mathcal{V}| \times |\mathcal{E}|}$$

where $s_k = 1$ if $e_k \in \mathcal{E}_j$ and $s_k = 0$ otherwise.

Note, therefore, that the local incidence matrix will contain columns of all zeros in correspondence to those edges not adjacent to v_j .

Proposition 1: Let $p^{(x,y)} \in \mathbb{R}^{|\mathcal{V}| \times 2}$ be the position vector for all agents. The rigidity matrix $R(p)$ can be defined as

$$R(p) = \begin{bmatrix} E_l(\mathcal{G}_1)^T & \dots & E_l(\mathcal{G}_{|\mathcal{V}|})^T \end{bmatrix} \left(I_{|\mathcal{V}|} \otimes p^{(x,y)} \right), \quad (4)$$

where $E_l(\mathcal{G}_i)$ is the local incidence matrix for node v_i .

This observation allows us to define the rigidity matrix in a way that separates the underlying graph and the actual position of each node. In the sequel, we explore the ramification of Proposition 1 via the introduction of a symmetric companion matrix, termed the *symmetric rigidity matrix*, and the notion of the *rigidity eigenvalue*.

B. The Rigidity Eigenvalue

Lemma 2.1 relates the property of infinitesimal rigidity for a given framework to the rank of a corresponding matrix. A main contribution of this work is the translation of the rank condition to that of a condition on the spectrum of a corresponding matrix that we term the *symmetric rigidity matrix*.

The symmetric rigidity matrix is a symmetric and positive-semidefinite matrix defined as

$$\mathcal{R} := R(p)^T R(p) \in \mathbb{R}^{2|\mathcal{V}| \times 2|\mathcal{V}|}. \quad (5)$$

An immediate consequence of the construction of the symmetric rigidity matrix is that $\mathbf{rk}[\mathcal{R}] = \mathbf{rk}[R(p)]$ [14], leading to the following corollary.

Corollary 2.2: A framework (\mathcal{G}, p) is infinitesimally rigid if and only if $\mathbf{rk}[\mathcal{R}] = 2|\mathcal{V}| - 3$.

The rank condition of Corollary 2.2 can be equivalently stated in terms of the eigenvalues of \mathcal{R} . Denoting the eigenvalues of \mathcal{R} as $\lambda_1 \leq \lambda_2 \leq \dots \leq \lambda_{2|\mathcal{V}|}$, note that infinitesimal rigidity is equivalent to $\lambda_1 = \lambda_2 = \lambda_3 = 0$ and $\lambda_4 > 0$. We will now show that, in fact, for any connected graph, the first three eigenvalues are always 0. This places an additional emphasis on the eigenvalue λ_4 as a measure of infinitesimal rigidity; consequently, we term λ_4 the *Rigidity Eigenvalue*.

The first result in this direction shows that the symmetric rigidity matrix is similar to a weighted Laplacian matrix.

Proposition 2: The symmetric rigidity matrix is similar to the weighted Laplacian matrix via a permutation of the rows and columns as

$$PRP^T = (I_2 \otimes E(\mathcal{G})) \begin{bmatrix} W_x & W_{xy} \\ W_{xy} & W_y \end{bmatrix} (I_2 \otimes E(\mathcal{G})^T), \quad (6)$$

where W_x , W_y , and W_{xy} are diagonal weighting matrices for each edge in \mathcal{G} such that for the edge $e_k = (v_i, v_j)$,

$$[W_x]_{kk} = (p_i^x - p_j^x)^2, \quad [W_y]_{kk} = (p_i^y - p_j^y)^2, \\ [W_{xy}]_{kk} = (p_i^x - p_j^x)(p_i^y - p_j^y),$$

and p_i^x (p_i^y) represents the x -coordinate (y -coordinate) of the position of agent i .

Proof: The proof is by direct construction using Proposition 1 and (5). Consider the permutation matrix P as

$$P = \begin{bmatrix} I_{|\mathcal{V}|} \otimes \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \\ I_{|\mathcal{V}|} \otimes \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \end{bmatrix}. \quad (7)$$

Then,

$$PRP^T = \begin{bmatrix} (I_{|\mathcal{V}|} \otimes p^{xT}) \hat{E} \hat{E}^T (I_{|\mathcal{V}|} \otimes p^x) & (I_{|\mathcal{V}|} \otimes p^{xT}) \hat{E} \hat{E}^T (I_{|\mathcal{V}|} \otimes p^y) \\ (I_{|\mathcal{V}|} \otimes p^{yT}) \hat{E} \hat{E}^T (I_{|\mathcal{V}|} \otimes p^x) & (I_{|\mathcal{V}|} \otimes p^{yT}) \hat{E} \hat{E}^T (I_{|\mathcal{V}|} \otimes p^y) \end{bmatrix},$$

where $\hat{E} = [E_l(\mathcal{G}_1)^T \dots E_l(\mathcal{G}_{|\mathcal{V}|})^T]$. Observe that each block matrix above is size $|\mathcal{V}| \times |\mathcal{V}|$. Furthermore, note that

$$(I_{|\mathcal{V}|} \otimes p_x^T) \hat{E} = E(\mathcal{G}) \underbrace{\begin{bmatrix} \ddots & & & \\ & (p_i^x - p_j^x) & & \\ & & \ddots & \\ & & & \ddots \end{bmatrix}},$$

a diagonal matrix of size $|\mathcal{E}| \times |\mathcal{E}|$

where $E(\mathcal{G})$ is simply the incidence matrix of the underlying graph \mathcal{G} (with arbitrary orientation). From this we are able to construct the diagonal weight matrices as stated in the proposition, concluding the proof. ■

The representation of the symmetric rigidity matrix as a weighted Laplacian allows for a more transparent understanding of certain eigenvalues related to this matrix. The next result shows that the first three eigenvalues of \mathcal{R} must equal zero for any connected graph \mathcal{G} .

Theorem 2.3: If the underlying graph \mathcal{G} in a framework is connected, then the symmetric rigidity matrix has at least three eigenvalues at the origin; that is, $\lambda_1 = \lambda_2 = \lambda_3 = 0$. Furthermore, the eigenvectors associated with each eigenvalue are respectively $v_1 = P^T [\mathbf{1}^T \quad \mathbf{0}^T]^T$, $v_2 = P^T [\mathbf{0}^T \quad \mathbf{1}^T]^T$, and $v_3 = P^T [(p^y)^T \quad -(p^x)^T]^T$, where P is defined in (7).

Proof: Recall that for any connected graph, one has $E(\mathcal{G})^T \mathbf{1} = 0$ [12]. Therefore, PRP^T must have two eigenvalues at the origin, with eigenvectors $u_1 = [\mathbf{1}^T \quad \mathbf{0}^T]^T$ and $u_2 = [\mathbf{0}^T \quad \mathbf{1}^T]^T$. To show that there is also a third eigenvalue at the origin, we constructively build another eigenvector corresponding to the zero eigenvalue that is linearly independent of u_1 and u_2 . Let $u_3 \in \mathbb{R}^{2|\mathcal{V}|}$ be the eigenvector corresponding to λ_3 , and construct it as

$$[u_3]_i = \begin{cases} 0, & i \in \{1, |\mathcal{V}| + 1\} \\ p_i^y - p_1^y, & i \leq |\mathcal{V}| \\ p_1^x - p_{i-|\mathcal{V}|}^x, & i > |\mathcal{V}| + 1 \end{cases}$$

Next, observe that $(I_2 \otimes E(\mathcal{G})^T)u_3 = [b_1^T \quad b_2^T]^T$ is such that b_1 is $\pm(p_i^y - p_1^y)$ only for edges incident to node v_i , and 0 otherwise. Similarly, b_2 is $\pm(p_1^x - p_i^x)$ only for edges incident to node v_i . It can now be verified that from this construction one has

$$\begin{bmatrix} W_x & W_{xy} \\ W_{xy} & W_y \end{bmatrix} (I_2 \otimes E(\mathcal{G})^T)u_3 = 0.$$

To conclude the proof, observe that $v_3 = u_3 + p_1^y u_1 - p_1^x u_2$ is a linear combination of the three eigenvectors. ■

Theorem 2.3 provides a precise characterization of the first three eigenvalues of the symmetric rigidity matrix. The explicit characterization of the corresponding eigenvectors is given for completeness. It should be noted, however, that the description of the eigenvectors might be of utility when exploring distributed strategies for computing the rigidity eigenvalue, analogously to, e.g., the case of the *distributed power iteration* developed for the connectivity eigenvalue in [33]. Theorem 2.3 can be used to arrive at the main result relating infinitesimal rigidity to the rigidity eigenvalue.

Theorem 2.4: A framework (\mathcal{G}, p) is infinitesimally rigid if and only if the rigidity eigenvalue is strictly positive, i.e. $\lambda_4 > 0$.

Proof: The proof is a direct consequence of Corollary 2.2 and Theorem 2.3. ■

Another useful observation relates infinitesimal rigidity of a framework to connectedness of the underlying graph.

Corollary 2.5: Rigidity of the framework (\mathcal{G}, p) implies connectedness of the graph \mathcal{G} .

Having established strong connections between the infinitesimal rigidity of a framework and spectral properties of the symmetric rigidity matrix, we can proceed to demonstrate how the rigidity eigenvalue can be used in a gradient control law for rigidity maintenance.

III. RIGIDITY MAINTENANCE

In a dynamic and uncertain environment, a team of mobile robots may not be able to maintain the same communication and sensing graph throughout the duration of a mission. Indeed, line of sight occlusions and sensing range limitations can cause links between agents to drop out. Furthermore, collisions with obstacles and among robots should be mandatorily avoided during motion. In order to take into account all these constraints, we use results from [27, 33] and propose the following definition of *neighboring agents*:

Definition 3.1: Two agents u and v are considered *neighbors* if and only if (i) their relative distance ℓ_{uv} is smaller than $D \in \mathbb{R}^+$ (the sensing range) and larger than $d_{\min} \in [0, D)$ (the safety range), (ii) their line of sight is not occluded by an obstacle, and (iii) neither u nor v are closer than d_{\min} to any other agent or obstacle.

Conditions (i) and (ii) are meant to take into account the typical sensing constraints in multi-robot applications (minimum/maximum communication and sensing range, occlusions), while the purpose of condition (iii) is to force disconnection from the group if an agent is colliding with any other agent or obstacle in the environment. By virtue of this definition, we are then able to embed into a unique *rigidity maintenance action* all the requirements stated above.¹

A way to encode the neighboring relationship of Definition 3.1 into the interaction graph \mathcal{G} is to suitably shape the weights $A_{uv} \geq 0$ of the edges joining neighboring agents. In particular, following [27], we define the weights A_{uv} as the product of four terms

$$A_{uv} = a_{uv} b_{uv} c_u c_v, \quad (8)$$

with the following properties: weights $a_{uv}(\ell_{uv}) \geq 0$ stay constant if $\ell_{uv} < d_0$, where $0 < d_0 < D$ is a desired inter-robot distance, and smoothly vanish as $\ell_{uv} \rightarrow D$ (Figure 2(a) shows an example for $d_0 = 8$ and $D = 16$). As for weights $b_{uv}(\ell_{uvo}) \geq 0$, let ℓ_{uvo} be the distance between the segment joining agents u and v and the closest obstacle point, and let $0 \leq d_{\min}^o < d_{\max}^o \leq D$ be a desired minimum and maximum link/obstacle distance. Then, $b_{uv}(\ell_{uvo})$ are designed to stay constant if $\ell_{uvo} \geq d_{\max}^o$ and to smoothly vanish as $\ell_{uvo} \rightarrow d_{\min}^o$ (Figure 2(b) shows an example for $d_{\min}^o = 0.3$ and $d_{\max}^o = 3$). Finally, weights $c_u(\ell_{uv}) \geq 0$ (respectively $c_v(\ell_{vu}) \geq 0$) are designed to stay constant if $\ell_{uv} \geq d_0$ for any other agent v , and to smoothly vanish as

¹In fact, whenever an agent will get too close to an obstacle or another agent, all its edges will disappear thus leading to a disconnected graph. Since infinitesimal rigidity implies rigidity which, in turn, implies connectivity, this will also cause loss of infinitesimal rigidity.

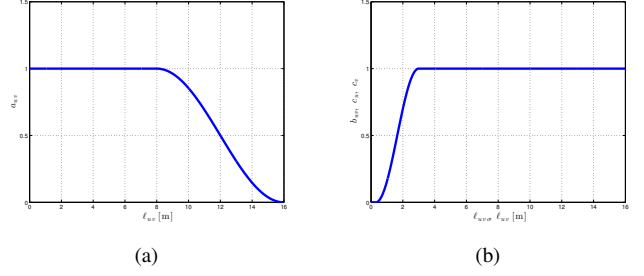


Fig. 2. Left: shape of the weights a_{uv} with $d_0 = 8$ and $D = 16$. Right: shape of the weights b_{uv} with $d_{\min}^o = 0.3$ and $d_{\max}^o = 3$. The shape of weights c_u and c_v is equivalent to that of b_{uv} .

$\ell_{uv} \rightarrow d_{\min}$ for any other agent v (see, again, Figure 2(b)). With these settings, the total weight A_{uv} (8) of the edge between agents u and v will vanish when any of the conditions of Definition 3.1 are not met.

We wish to stress that the weights A_{uv} are function only of the *relative distances* among agents and *relative distances* among the segment joining two agents and the closest obstacles. Therefore, this is consistent with the assumption of relying solely on relative measurements for the purpose of formation control; see the discussion in the Introduction.

A. Gradient of the Rigidity Eigenvalue

In order to use the rigidity eigenvalue as a feedback mechanism for rigidity maintenance, it is first required that an expression for its gradient with respect to the positions of each agent is determined. In this subsection, we compute this gradient and demonstrate that it can in fact be described entirely by *relative positions* of each agent in the formation.

In this direction, first recall that the rigidity eigenvalue can be expressed as

$$\lambda_4 = v_4^T P R P^T v_4,$$

where v_4 is the *normalized* eigenvector associated with λ_4 , which we call the *rigidity eigenvector*. For notational convenience, we denote $v_4 = [(v^x)^T \ (v^y)^T]^T$, and we index each element of this vector with a subscript. Expanding this expression, we can obtain

$$\lambda_4 = \left[\left(\sum_{i \sim j} (p_i^x - p_j^x)^2 (v_i^x - v_j^x)^2 \right) + \left(\sum_{i \sim j} (p_i^y - p_j^y)^2 (v_i^y - v_j^y)^2 \right) + \left(2 \sum_{i \sim j} (p_i^x - p_j^x)(p_i^y - p_j^y)(v_i^x - v_j^x)(v_i^y - v_j^y) \right) \right].$$

The above expression lends itself to a straightforward calculation of the gradient with respect to each agent's position.

$$\frac{\partial \lambda_4}{\partial p_i^x} = 2 \left(\sum_{i \sim j} (p_i^x - p_j^x)(v_i^x - v_j^x)^2 + (p_i^y - p_j^y)(v_i^x - v_j^x)(v_i^y - v_j^y) \right) \quad (9)$$

$$\frac{\partial \lambda_4}{\partial p_i^y} = 2 \left(\sum_{i \sim j} (p_i^y - p_j^y)(v_i^y - v_j^y)^2 + (p_i^x - p_j^x)(v_i^x - v_j^x)(v_i^y - v_j^y) \right). \quad (10)$$

The key feature of (9) and (10) is that the gradients are expressed entirely in terms of the relative positions of neighboring pairs. It is important to emphasize here that, despite this distributed structure, each agent must have access to some global information, namely, the relative components of the rigidity eigenvector.

B. Gradient of the Rigidity Eigenvalue with Weighted Links

The previous derivation of the gradient of the rigidity eigenvalue can be easily extended to the case of weighted links, that is, the case under consideration in this paper. To this end, we redefine the elements of matrixes W_x , W_y , W_{xy} in (6) as

$$[W_x]_{kk} = (p_i^x - p_j^x)^2 A_{ij}, \quad [W_y]_{kk} = (p_i^y - p_j^y)^2 A_{ij}, \\ [W_{xy}]_{kk} = (p_i^x - p_j^x)(p_i^y - p_j^y) A_{ij}.$$

where A_{ij} are the weights defined in (8). With this formulation, the weighted rigidity eigenvalue can now be expressed as

$$\lambda_4 = \left(\sum_{i \sim j} (p_i^x - p_j^x)^2 A_{ij} (v_i^x - v_j^x)^2 \right) + \left(\sum_{i \sim j} (p_i^y - p_j^y)^2 A_{ij} (v_i^y - v_j^y)^2 \right) + \left(2 \sum_{i \sim j} (p_i^x - p_j^x)(p_i^y - p_j^y) A_{ij} (v_i^x - v_j^x)(v_i^y - v_j^y) \right).$$

It can be verified that the computation of the gradient of λ_4 w.r.t. the pair (p_i^x, p_i^y) is a straightforward extension of the previous case. In fact,

$$\frac{\partial \lambda_4}{\partial p_i^x} = \sum_{i \sim j} \left(2(p_i^x - p_j^x) A_{ij} (v_i^x - v_j^x)^2 + (p_i^x - p_j^x)^2 \frac{\partial A_{ij}}{\partial p_i^x} (v_i^x - v_j^x)^2 + 2(p_i^y - p_j^y) A_{ij} (v_i^x - v_j^x)(v_i^y - v_j^y) + 2(p_i^x - p_j^x)(p_i^y - p_j^y) \frac{\partial A_{ij}}{\partial p_i^x} (v_i^x - v_j^x)(v_i^y - v_j^y) \right) \quad (11)$$

and analogously for $\frac{\partial \lambda_4}{\partial p_i^y}$.

C. The Rigidity Potential

As a final step, we define a scalar potential function $V_\lambda(\lambda_4) > 0$ with the properties of growing unbounded as $\lambda_4 \rightarrow \lambda_4^{\min} > 0$ and vanishing (with vanishing derivative) as $\lambda_4 \rightarrow \infty$. A possible shape for V_λ is illustrated in Figure 3 with $\lambda_4^{\min} = 5$. In order to maintain infinitesimal rigidity of

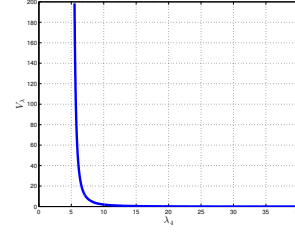


Fig. 3. A possible shape for the rigidity potential function V_λ with $\lambda_4^{\min} = 5$.

the formation (while adopting the neighboring definition in Definition 3.1), it is then sufficient that every agent follows the anti-gradient of V_λ , that is, applies the control

$$\xi_i = -\frac{\partial V_\lambda}{\partial p_i} = -\frac{\partial V_\lambda}{\partial \lambda_4} \frac{\partial \lambda_4}{\partial p_i}. \quad (12)$$

where ξ_i is the agent velocity input defined in (3), and $p_i = [p_i^x \ p_i^y]^T$ is the position vector of the i th agent. We note, again, that the agent control action (12) is *almost* decentralized, since it is a function of *relative positions* among neighbors and closest obstacles. The only pieces of global information required in (12) are the current value of λ_4 (needed for evaluating $\frac{\partial V_\lambda}{\partial \lambda_4}$), and the current entries of the

eigenvector v_4 (needed for evaluating $\frac{\partial \lambda_4}{\partial p_i}$). We are currently working towards a decentralized algorithm for estimating λ_4 and v_4 online, similar to what has been done for the case of the connectivity eigenvalue and eigenvector obtained from the graph Laplacian matrix, see, e.g., [33].

As a final remark, note that the control action (12), while ensuring rigidity maintenance for graph \mathcal{G} and, implicitly, inter-robot and obstacle collision avoidance, *will not prevent* the creation or disconnection of individual links (as per Definition 3.1). Therefore, the graph topology will be allowed to change over time whenever needed, for example because of sensing limitations or autonomous split/join decisions. This constitutes an additional feature of our approach, as the robots will not be overly constrained, e.g., by requiring maintenance of a *given* fixed graph topology.

IV. SIMULATIONS

In this last Section we report the simulation results of a case study involving 7 quadrotor UAVs exploited as mobile robotic platforms. Assuming availability of standard trajectory tracking controllers [6, 16], we model the closed-loop quadrotor behavior as a single integrator in \mathbb{R}^3 , see also [10, 11, 28] for similar working assumptions. As explained in Section II-A, here, for the sake of illustration, we consider tridimensional agents (and related tridimensional version of the rigidity maintenance controller (12)), although the previous developments have been worked out only for the planar case. In fact, extension of the proposed machinery to \mathbb{R}^3 is a straightforward exercise.

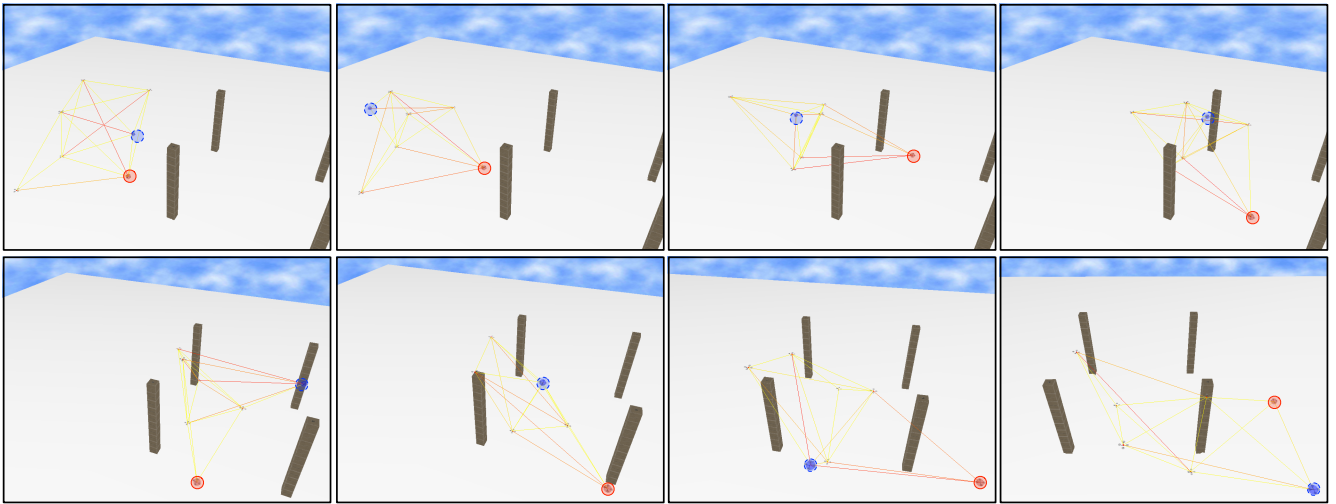


Fig. 4. Snapshots of a simulation with 7 quadrotor UAVs. The two UAVs partially controlled by two human operators are marked with two semi-transparent spheres.

In order to better illustrate the overall behavior of the system (maintenance of infinitesimal rigidity despite the time-varying topology due to the neighboring conditions of Definition 3.1), we added an exogenous bounded term $u_i^* \in \mathbb{R}^3$ to the control action (12) for two of the 7 quadrotors, namely, quadrotors 1 and 2. This way, two human operators could independently add two velocity commands to quadrotors 1 and 2 by acting on two joysticks during the simulation. Being that the commands u_i^* are bounded, their effect does not threaten rigidity maintenance since, intuitively, the feedback term (12) is eventually always dominant as the slope of V_λ grows unbounded when approaching loss of infinitesimal rigidity. Figure 4 shows eight snapshots taken during the simulation; a video of this simulation run can be found at <http://youtu.be/Ni6rIrcA5Hw>. The quadrotors controlled by the two human operators are marked with two semi-transparent spheres, while the links joining any two agents change brightness from light (yellow) to dark (red) as the weights A_{uv} drop from their maximum value to zero. The quadrotors were flown in the cluttered environment shown in Figure 4 with the aim of triggering as much as possible split and join events because of the constraints of Definition 3.1, namely, maximum communication/sensing range, line-of-sight occlusion, obstacle and inter-robot minimum distance.

Figure 5(a) reports the behavior of the rigidity eigenvalue² during the simulation: one can appreciate that, being positive at all times, infinitesimal rigidity is never compromised despite the complex maneuvering among the obstacles. Furthermore, we show in Figure 5(b) the behavior of the connectivity eigenvalue, the eigenvalue associated to the connectivity of the graph \mathcal{G} . This is meant to show that, as expected from Corollary 2.5, connectivity of the graph is also automatically preserved when ensuring graph rigidity. Finally, Figure 6 reports the total number of edges $|\mathcal{E}|$ of the graph \mathcal{G} over time

²Note that for $d = 3$, the rigidity eigenvalue is λ_7 of the symmetric rigidity matrix.

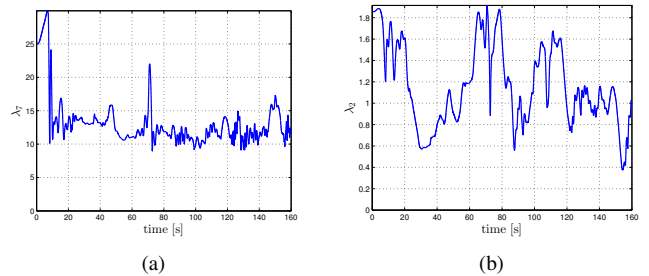


Fig. 5. A plot of the rigidity eigenvalue and connectivity eigenvalue of the simulation. Note how both keep positive at all times, confirming that infinitesimal rigidity and, as a consequence, connectivity of the graph were always maintained.

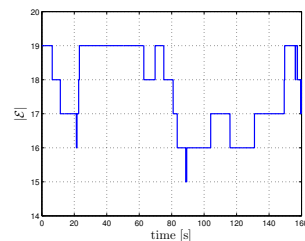


Fig. 6. The total number of edges $|\mathcal{E}|$ over time. Note how the graph topology changes over time as edges are created and disconnected.

in order to show that the graph topology is actually varying because of the frequent split and join events.

V. CONCLUSION

This work presented a gradient controller for the maintenance of infinitesimal rigidity in a multi-robot team. A key component of this work was the identification of an important parameter for ensuring infinitesimal rigidity that we termed the rigidity eigenvalue. We provided necessary and sufficient conditions that relate the positivity of the rigidity eigenvalue

to the infinitesimal rigidity of the formation. This, in turn, was used to develop a gradient controller for rigidity maintenance which was shown to have a distributed structure depending only on the relative positions of each robot and relative values of the eigenvector associated with the rigidity eigenvalue. We also augmented this controller with obstacle and collision avoidance features. The results were demonstrated with a simulation example. An important extension of this work that we are considering is the distributed determination of the rigidity eigenvalue and eigenvector.

ACKNOWLEDGMENTS

This research was partly supported by WCU (World Class University) program funded by the Ministry of Education, Science and Technology through the National Research Foundation of Korea (R31-10008).

REFERENCES

- [1] I. F. Akyildiz, Y. Sankarasubramaniam, and E. Cayirci. A survey on sensor networks. *IEEE Communications Magazine*, 40(8):102–114, 2002.
- [2] B. D. O. Anderson, B. Fidan, C. Yu, and D. van der Walle. UAV formation control: Theory and application. In V. D. Blondel, S. P. Boyd, and H. Kimura, editors, *Recent Advances in Learning and Control*, volume 371 of *Lecture Notes in Control and Information Sciences*, pages 15–34. Springer, 2008.
- [3] B. D. O. Anderson, C. Yu, B. Fidan, and J. M. Hendrickx. Rigid graph control architectures for autonomous formations. *IEEE Control Systems Magazine*, 28(6):48–63, 2008.
- [4] J. Aspnes, T. Eren, D. K. Goldenberg, A. S. Morse, W. Whiteley, Y. R. Yang, B. D. O. Anderson, and P. N. Belhumeur. A theory of network localization. *IEEE Trans. on Mobile Computing*, 5(12):1663–1678, 2006.
- [5] J. Baillieul and L. McCoy. The combinatorial graph theory of structured formations. In *2007 46th IEEE Conference on Decision and Control*, pages 3609–3615, December 2007.
- [6] S. Bouabdallah and R. Siegwart. Backstepping and sliding-mode techniques applied to an indoor micro. In *2005 IEEE Int. Conf. on Robotics and Automation*, pages 2247–2252, May 2005.
- [7] J. Bristow, D. Folta, and K. Hartman. A Formation Flying Technology Vision. In *AIAA Space 2000 Conference and Exposition*, volume 21, Long Beach, CA, April 2000.
- [8] G. C. Calafiore, L. Carlone, and M. Wei. A distributed gradient method for localization of formations using relative range measurements. In *2010 IEEE Int. Symp. on Computer-Aided Control System Design*, pages 1146–1151, Yokohama, Japan, Sep. 2010.
- [9] R. Connelly and W.J. Whiteley. Global Rigidity: The Effect of Coning. *Discrete Computational Geometry*, 43(4):717–735, 2009.
- [10] J. Fink, N. Michael, S. Kim, and V. Kumar. Planning and control for cooperative manipulation and transportation with aerial robots. In *14th Int. Symp. on Robotics Research*, Lucerne, Switzerland, Sep. 2009.
- [11] A. Franchi, C. Secchi, M. Ryll, H. H. Bühlhoff, and P. Robuffo Giordano. Shared control: Balancing autonomy and human assistance with a group of quadrotor UAVs. *IEEE Robotics & Automation Magazine, Special Issue on Aerial Robotics and the Quadrotor Platform*, 19(3):57–68, 2012.
- [12] C. D. Godsil and G. Royle. *Algebraic Graph Theory*. Springer, 2001. ISBN 978-0-387-95241-3.
- [13] J. Graver, B. Servatius, and H. Servatius. *Combinatorial Rigidity*. Graduate Studies in Mathematics. American Mathematical Society, Providence, RI, 1993. ISBN 0821838016.
- [14] R. Horn and C Johnson. *Matrix Analysis*. Cambridge University Press, New York, 1985.
- [15] R.A. Horn and C.R. Johnson. *Topics in Matrix Analysis*. Cambridge University Press, New York, NY, 1991.
- [16] M.-D. Hua, T. Hamel, P. Morin, and C. Samson. A control approach for thrust-propelled underactuated vehicles and its application to VTOL drones. *IEEE Trans. on Automatic Control*, 54(8):1837–1853, 2009.
- [17] B. Jackson. Notes on the Rigidity of Graphs. In *Levico Conference Notes*, 2007.
- [18] D. Jacobs. An Algorithm for Two-Dimensional Rigidity Percolation: The Pebble Game. *Journal of Computational Physics*, 137(2):346–365, November 1997.
- [19] Y. Kim, G. Zhu, and J. Hu. Optimizing formation rigidity under connectivity constraints. In *49th IEEE Conf. on Decision and Control*, pages 6590–6595, Atlanta, GA, Dec. 2010.
- [20] L. Krick, M. E. Broucke, and B. A. Francis. Stabilisation of infinitesimally rigid formations of multi-robot networks. *International Journal of Control*, 82(3):423439, 2009.
- [21] G. Laman. On graphs and rigidity of plane skeletal structures. *Journal of Engineering Mathematics*, 4(4):331–340, 1970.
- [22] Q. Lindsey, D. Mellinger, and V. Kumar. Construction of cubic structures with quadrotor teams. In *2011 Robotics: Science and Systems*, Los Angeles, CA, Jun. 2011.
- [23] M. Mesbahi and M. Egerstedt. *Graph Theoretic Methods in Multiagent Networks*. Princeton Series in Applied Mathematics. Princeton University Press, 1 edition, 2010. ISBN 9780691140612.
- [24] N. Michael, J. Fink, and V. Kumar. Cooperative manipulation and transportation with aerial robots. In *2009 Robotics: Science and Systems*, Seattle, WA, Jun. 2009.
- [25] R. M. Murray. Recent research in cooperative control of multi-vehicle systems. *ASME Journal on Dynamic Systems, Measurement, and Control*, 129(5):571–583, 2006.
- [26] R. Olfati-Saber and R. M. Murray. The combinatorial graph theory of structured formations. In *41th IEEE Conf. on Decision and Control*, pages 3609–3615, Las Vegas, NV, Dec. 2002.
- [27] P. Robuffo Giordano, A. Franchi, C. Secchi, and H. H. Bühlhoff. Passivity-based decentralized connectivity maintenance in the bilateral teleoperation of multiple UAVs. In *2011 Robotics: Science and Systems*, Los Angeles, CA, Jun. 2011.
- [28] M. Schwager, B. Julian, M. Angermann, and D. Rus. Eyes in the sky: Decentralized control for the deployment of robotic camera networks. *Proceedings of the IEEE*, 99(9):1541–1561, 2011.
- [29] I. Shames, B. Fidan, and B. D. O. Anderson. Minimization of the effect of noisy measurements on localization of multi-agent autonomous formations. *Automatica*, 45(4):1058–1065, 2009.
- [30] B. Smith, M. Egerstedt, and A. Howard. Automatic generation of persistent formations for multi-agent networks under range constraints. In *1st Int. Conf. on Robot communication and coordination*, pages 1–8, 2007.
- [31] T. Tay and W. Whiteley. Generating isostatic frameworks. *Structural Topology*, 11(1):21–69, 1985.
- [32] C. Wu, Y. Zhang, W. Sheng, and S. Kanchi. Rigidity guided localisation for mobile robotic sensor networks. *International Journal of Ad Hoc and Ubiquitous Computing*, 6(2):114, 2010.
- [33] P. Yang, R. A. Freeman, G. J. Gordon, K. M. Lynch, S. S. Srinivasa, and R. Sukthankar. Decentralized estimation and control of graph connectivity for mobile sensor networks. *Automatica*, 46(2):390–396, 2010.
- [34] G. Zhu and J. Hu. Stiffness matrix and quantitative measure of formation rigidity. In *48th IEEE Conf. on Decision and Control*, pages 3057–3062, Shanghai, China, Dec. 2009.