# Graph Rigidity and Distributed Formation Stabilization of Multi-Vehicle Systems

Reza Olfati-Saber    Richard M. Murray
Control and Dynamical Systems
California Institute of Technology
Pasadena, CA 91125
e-mail: {olfati,murray}@cds.caltech.edu

## Abstract

In this paper, we provide a graph theoretical framework that allows us to formally define formations of multiple vehicles and the issues arising in uniqueness of graph realizations and its connection to stability of formations. The notion of graph rigidity is crucial in identifying the shape variables of a formation and an appropriate potential function associated with the formation. This allows formulation of meaningful optimization or nonlinear control problems for formation stabilization/tacking, in addition to formal representation of split, rejoin, and reconfiguration maneuvers for multi-vehicle formations. We introduce an algebra that consists of performing some basic operations on graphs which allow creation of larger rigid-by-construction graphs by combining smaller rigid subgraphs. This is particularly useful in performing and representing rejoin/split maneuvers of multiple formations in a distributed fashion.

## 1 Introduction

Coordinated control of multi-agent/multi-vehicle systems in a distributed fashion has attracted several researchers with rather diverse backgrounds in control theory, computer science, biology, and physics. Multi-agent systems arise in broad areas including formation flight of unmanned aerial vehicles (UAVs), coordination of satellite clusters, automated highways, understanding the coordination and movement of flocks of birds or schools of fish [1], and molecular conformation problems.

The applications that are of primary interest in our work include performing maneuvers by UAVs which (possibly) require doing split/rejoin maneuvers in case a group of vehicles come across an obstacle (see Fig. 1). In addition, we are interested in reconfiguration of the formation of a group of vehicles (Fig. 2) due to a change of the team-strategy in team-on-team competitive games.
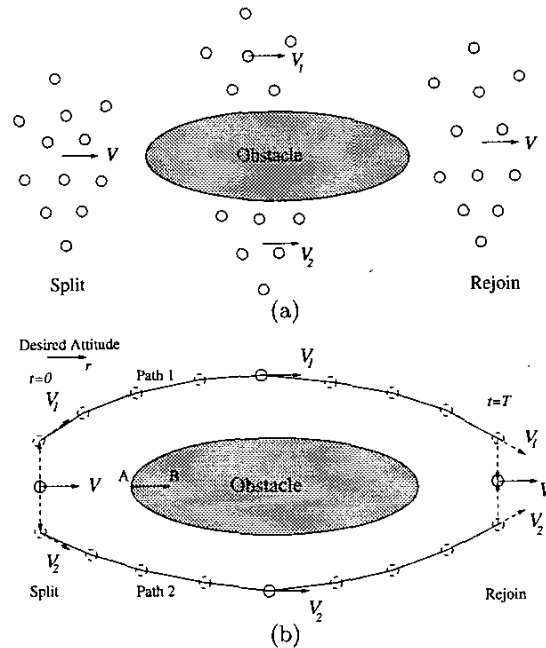


Figure 1: (a) Split/Rejoin maneuver for a group of 12 vehicles moving in $\mathbb{R}^2$ and (b) the paths of centers of mass of two subgroups of vehicles in split/rejoin maneuver for a group of 12 vehicles moving in $\mathbb{R}^2$.

As a result, our theoretical objective is to provide analytical and computational tools for representation and manipulation of formations of multiple vehicles such as performing split, rejoin, and reconfiguration maneuvers in a distributed manner. As we realized from one of our earlier works [2], a notion from graph theory called *graph rigidity* turns out to be instrumental in both representation and distributed coordinated control of formations of multiple vehicles. *Minimally rigid graphs* (i.e. rigid graphs with $n$ nodes and $2n-3$ directed edges, see section 2.3) are an important class of rigid graphs that their edges are closely related to shape variables of formations of $n$ vehicles. This in

turn leads to automatic generation of potential functions from the interconnection graph of the group of vehicles that guarantee local *structural formation stabilization* [2].

One of the contributions of this paper is introducing new properties of minimally rigid graphs that allow composition of smaller rigid subgraphs that construct a larger rigid graph. This is used to represent and perform rejoin/split maneuvers for groups of vehicles. We introduce an *algebra over graphs* that allows performing some basic operations on graphs including rejoining two graphs, node augmentation to a graph, and attaching graphs via their edges.
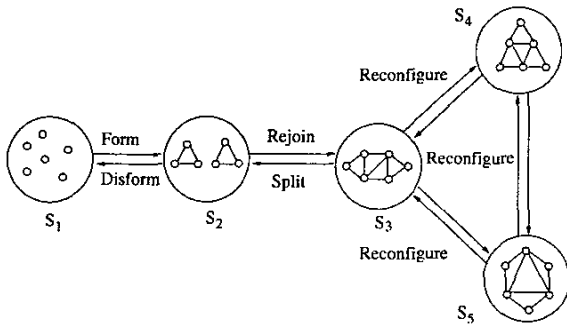


**Figure 2:** A Hybrid System representing the switching between multiple formations via performing a set of maneuvers.

In general, performing several types of operations on formations might be necessary. For example, performing split/rejoin maneuvers, shown in Fig. 1 (a) (see section 3 for a detailed discussion). A *Hybrid System* (Fig. 2) can be used to provide a high-level representation of a sequence of admissible operations on formations through performing certain maneuvers as discussed earlier. In Fig. 2, a diagram of such a hybrid system is presented. Each discrete-state of this hybrid systems consists of *rigid formation graphs* that represent the information flow among a group of vehicles.

In the past, several methodologies have been exploited by different researchers in distributed control and coordination of multi-vehicle systems that use a graph theoretical framework [3, 4, 2]. Furthermore, the work in [5] identifies connections between graph rigidity and multi-vehicle formations without providing any control design approaches for stabilization of such formations based on rigidity.

The field of graph rigidity is very broad and several scientists and engineers from rather diverse backgrounds in mathematics, physics, chemistry, biology, computer science, and mechanical and civil engineering have been actively working on this subject over the past three decades (see [6] for a complete survey and the history of

the subject that goes back to Euler in 1766 and Cauchy in 1813). Here, we are interested in *combinatorial rigidity* [7, 8] as supposed to *infinitesimal rigidity* [9].

This paper is organized as follows. The mathematical preliminaries required to discuss multi-agent formations, graph rigidity, and minimally rigid graphs are presented in Section 2. Our main results on split/rejoin of groups of agents are presented in Section 3. Finally, concluding remarks are given in Section 4.

## 2 Preliminaries: Formations and Graph Rigidity

In this section, we provide the mathematical preliminaries that allow us to define formations of multiple agents/vehicles and its connection to graph rigidity.

### 2.1 Formations of Multi-Vehicles
In this section, we define a formation of $n$-agents, the position, and the attitude of a formation. Consider a group of $n$ vehicles ($n \geq 2$) each with the following dynamics

$$\begin{cases} \dot{q}_i &= p_i \\ \dot{p}_i &= u_i \end{cases} \tag{1}$$

where $q_i, p_i, u_i \in \mathbb{R}^m$ for all $i \in \mathcal{I} = \{1, \ldots, n\}$. Therefore, each vehicle has a *linear dynamics*.
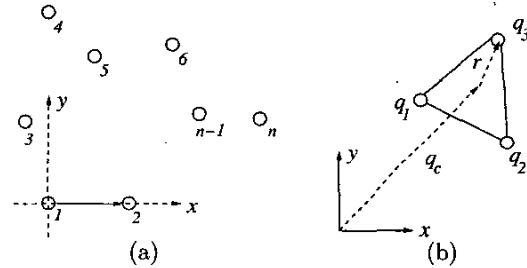


**Figure 3:** (a) A formation of $n \geq 2$ agents with a base $(1, 2)$ in $\mathbb{R}^2$, and (b) position and attitude $(q_c, r)$ of the formation of three vehicles.

We refer to a set of $n$ points in $\mathbb{R}^m$ as an *$n$-grid*. The column vector $q = (q_1, \ldots, q_n)^c \in \mathbb{R}^{mn}$ is called the *configuration* of the $n$-grid. Identifying an agent $i \in \mathcal{I}$ by its position $q_i$, an agent can be viewed as a point in $\mathbb{R}^m$. Assume $\|q_2 - q_1\| > 0$ and connect the agents 1 and 2 by a directed partial-line $e_{12}$ that is called the *base-edge* of the $n$-grid. An $n$-grid in which the distance between each two agents is greater than zero is called *collision-free*. The vector $e_{12}$ and its orthogonal $e_{12}^{\perp}$ (after $+\pi/2$ rotation) determine a body axes for the n-grid. Let $(x_i, y_i), 3 \leq i \leq n$ denote the coordinates of the $i$th agent in the body axes (see Fig. 3 (a)). Then, an $n$-grid is uniquely specified by the following $(2n-3)$-dimensional vector

$$\varphi = (l, x_3, y_3, x_4, y_4, \ldots, x_n, y_n) \in Q := \mathbb{R}_{\geq 0} \times \mathbb{R}^{2n-4} \tag{2}$$

We refer to $\varphi$ as the vector of *internal degrees of freedom* of an $n$-grid. Apparently, $\varphi$ remains invariant under rotation and translation of all the points in an $n$-grid. Let $q_c = 1/n \sum_{i \in \mathcal{I}} q_i$ and $\mathbf{r} = \mathbf{n}(q_{j^*} - q_c)$ where $j^* \in \mathcal{I}$ is called the *attitude leader* and $\mathbf{n}(x) = x/\|x\|$ denotes the normalization of vector $x \neq 0$. Then, $q_c$ and $\mathbf{r}$ are called the *position and attitude* of an $n$-grid (see Fig. 3 (b)) which determine the *external degrees of freedom* of an $n$-grid.

*Definition 1.* (formation) A *formation* $\varphi$ of $n$-agents is a point on the manifold $Q$ (defined in (2)) associated with the set of all $n$-grids in $\mathbb{R}^2$ that have the same internal degrees of freedom. The *position and attitude* of a formation is defined as $\psi = (q_c, \mathbf{r})$.

The method that is introduced here for representation of formations of planar $n$-grids can be directly generalized to any other dimension $m \geq 3$.

## 2.2 Graph Rigidity
Let $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathcal{W})$ be a weighted graph with the set of vertices $\mathcal{V} = \{v_1, \ldots, v_n\}$ (i.e. $|\mathcal{V}| = n$), the set of edges $\mathcal{E}$, and the set of weights $\mathcal{W}$. In addition, define $\mathcal{I} = \{1, 2, \ldots, n\}$ as the set of indices of the element of $\mathcal{V}$. Each agent in a multi-agent system can be viewed as a node of the graph $\mathcal{G}$ which represents the overall system.

*Remark 1.* Throughout this paper, we assume that controller of the multi-agent system is *distributed*. This means that each agent performs *sensing and communication* with all of its *neighbors* $J_i := \{j \in \mathcal{I} : e_{ij} \in \mathcal{E}\}$ in a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$. As a special case, this definition of a neighbor includes the case of *spatial neighbors* of an agent that are located within a distance $d > 0$ of each agent (see [1]).

Let $q_i \in \mathbb{R}^m$ denote the coordinates vector assigned to node $v_i$ of the graph. Then $q = \mathrm{col}(q_1, \cdots, q_n) \in \mathbb{R}^{mn}$ is called a *realization* of $\mathcal{G}$ iff

$$\|q_j - q_i\| = w_{ij}, \quad \forall e_{ij} \in \mathcal{E}, q_i, q_j \in \mathbb{R}^m$$

where $\mathcal{W} = \{w_{ij}\}, \mathcal{E} = \{e_{ij}\}$. The pair $(\mathcal{G}, q)$ is called a *framework*. An *infinitesimal motion* is an assignment of a velocity vector $p_i$ to the vertex $v_i$ of the graph $\mathcal{G}$ such that

$$\langle p_j - p_i, q_j - q_i \rangle = 0, \quad \forall e_{ij} \in \mathcal{E} \tag{3}$$

where $\langle \cdot, \cdot \rangle$ denotes the inner product. A *flexing* of a framework $(\mathcal{G}, q)$ is a family of realizations of $\mathcal{G}$ parameterized by $t$, $q(t) : [0, 1] \to \mathbb{R}^{mn}$ such that $q(0) = q$ and $q(t)$ is a differentiable function of $t$. Let $p(t) = \dot{q}$ be the vector of velocities assigned to each node. Then, a flexing $q(t)$ satisfies the following relation

$$\langle p_j(t) - p_i(t), q_j(t) - q_i(t) \rangle = 0, \quad \forall e_{ij} \in \mathcal{E}, \forall t \in [0, 1] \tag{4}$$

Clearly, the rigid motions of $\mathbb{R}^m$ are length preserving and $p_i = p_i(t)$ defines an infinitesimal motion of the family of graphs $\mathcal{G}(t)$ with realization $q(t)$ where $t \in [0, 1]$. These rigid motions are called *trivial* flexings of a framework. A framework $(\mathcal{G}, q)$ is called *infinitesimally rigid* iff the only infinitesimal motions of the framework are trivial flexings.

It turns out that rigidity of graphs is a generic property in the sense that almost all realizations of a particular graph are either infinitesimally rigid, or flexible [10]. Thus, rigidity is a *generic* property of graphs. This eliminates the necessity to check the rigidity property for all possible realizations of a graph.

*Note.* In the context of *rigidity* of graphs, it is assumed that *all the edges are directed* and the graph has neither undirected edges, nor edges from one node to itself.

Throughout this paper, we assume each agent performs *sensing and communication* with all of its neighbors that are defined as the following:

*Definition 2.* (neighbors) The indices of the *neighbors* of the node $v_i$ in the graph $\mathcal{G} = (V, \mathcal{E})$ is denoted by $J_i$ and defined as $J_i := \{j \in \mathcal{I} : e_{ij} \in \mathcal{E}\}$.

A combinatorial characterization of rigidity of graphs in $\mathbb{R}^2$ was first obtained by Laman [7]. First, we need to define the Laman subgraph of a graph.

*Definition 3.* (Laman subgraph) A *Laman subgraph* of a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ is a graph $\mathcal{H} = (\mathcal{V}_\mathcal{H}, \mathcal{E}_\mathcal{H})$ such that $\mathcal{V}_\mathcal{H} \subseteq \mathcal{V}$, $|\mathcal{V}_\mathcal{H}| \geq 2$, and $\mathcal{E}_\mathcal{H} = \mathcal{E}|_{\mathcal{V}_\mathcal{H}} := \{e_{ij} \in \mathcal{E} : v_i, v_j \in \mathcal{V}_\mathcal{H}\}$ (We read $\mathcal{E}_\mathcal{H}$ is the restriction of $\mathcal{E}$ to $\mathcal{V}_\mathcal{H}$)

**Theorem 1.** *(Laman, 1970 [7]) A planar graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ with $n \geq 2$ nodes is rigid iff there exists a subset $\mathcal{E}_\mathcal{H} \subseteq \mathcal{E}$ with $2n - 3$ edges of $\mathcal{G}$ such that for the graph $\mathcal{H} = (\mathcal{V}, \mathcal{E}_\mathcal{H})$ with $n$ nodes, each Laman subgraph $Y = (\mathcal{V}_Y, \mathcal{E}_Y)$ of $\mathcal{H}$ satisfies the property $|\mathcal{E}_Y| \leq 2|\mathcal{V}_Y| - 3$.*

We refer to $\mathcal{H}$ in Theorem 1 as the *essential subgraph* of the rigid graph $\mathcal{G}$. An edge $e$ of a rigid graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ is called *redundant*, iff after removing $e$ from $\mathcal{E}$, the graph remains rigid. Otherwise, $e$ is called an *independent* edge.

According to Laman's Theorem, *any planar rigid graph $\mathcal{G}$ ($n \geq 2$) has at least $2n - 3$ edges*. Laman's Theorem was later generalized by Lovász and Yemini in [8]. Based on their work, in [11] an $O(n^2)$ algorithm is available that determines whether a graph is rigid or not.

## 2.3 Minimally Rigid Graphs
Any rigid graph $\mathcal{G}$ with $n \geq 2$ nodes and $2n - 3$ edges is called a *minimally rigid graph* (MRG). Apparently, any MRG is *the essential subgraph of itself*. In addition, *every edge of an MRG is independent*.

Due to computational and communications costs in a

network of $n$-vehicles, we are interested in the least possible number of edges between the agents that creates a rigid graph and thus a locally stabilizing distributed control law for each vehicle [2]. This makes minimally rigid graphs the ideal choice for us. Moreover, it will become clear later that MRGs benefit from nice analytic properties that allow one to construct bigger graphs through connecting minimally rigid subgraphs. This is explained in complete details in section 3.

The edges of a minimally rigid graph $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathcal{W})$ define the following set of *shape variables* for the graph:

$$\eta_{ij} := \|q_j - q_i\| - w_{ij}, \quad \forall e_{ij} \in \mathcal{E} \qquad (5)$$

We call the column vector $\eta$ and manifold $Q(\mathcal{G})$ defined by

$$\eta = \{\eta_{ij}\} \in Q(\mathcal{G}) := \Pi_{e_{ij} \in \mathcal{E}}[-w_{ij}, \infty) \subset \mathbb{R}^{2n-3} \qquad (6)$$

the *shape configuration* and *shape manifold* of $\mathcal{G}$. Any point at the boundary of $Q(\mathcal{G})$ *corresponds to a collision between two agents.* The *structural potential function* of the graph $\mathcal{G}$ is defined as a smooth, proper, and positive definite function $V(\eta)$ that satisfies $V(0) = 0$. Two examples of $V(\eta)$ (or $V(q)$) are given in [2] as the following:

$$\begin{array}{rcl} V_1(\eta) & = & \sum_{e_{ij} \in \mathcal{E}} \eta_{ij}^2 \\ V_2(\eta) & = & \sum_{e_{ij} \in \mathcal{E}}[(1 + \eta_{ij}^2)^{\frac{1}{2}} - 1] \end{array} \qquad (7)$$

Clearly, $V_2(q) := V_2(\eta)$ has a bounded gradient w.r.t. $q$ and this is the key in designing a bounded control input for structural formation stabilization [2].

## 3 Main Results: Rejoin/Split of Formations of Multi-Agents

To study and manipulate rejoin and split of formations of multiple agents, we need to formally define some basic operations on graphs that preserve the rigidity properties of the obtained graphs. For doing so, we discuss the problem of joining two rigid graphs to construct a new composite rigid graph. This operation is called *rejoining graphs.* Later, we discuss how a rigid graph can be decomposed into two sets of disjoint graphs that are both rigid. The second operation is called *splitting a graph.* These basic operations (together with some other operations) constitute a novel *graph algebra.* This graph algebra greatly facilitates representation of the rejoin/split operations on graphs.

### 3.1 Node Augmentation to Graphs
A key ingredient in formal definition of rejoin/split of rigid graphs is the *node augmentation* operation. Before, we describe this operation, we need to define proper Laman subgraphs and present an axiom.

*Definition* 4. (proper Laman subgraphs) A Laman subgraph $Y = (\mathcal{V}_Y, \mathcal{E}_Y)$ of a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ satisfying

the property $|\mathcal{E}_Y| \leq 2|\mathcal{V}_Y| - 3$ is called a *proper Laman subgraph.*

**Axiom 1.** *(single node rigidity) Any graph with a single node is minimally rigid in $\mathbb{R}^m$.*

*Definition* 5. (node augmentation for single node graphs) Consider two single node graphs $\mathcal{G}_1 = (\{v_1\}, \emptyset, \emptyset)$ $\mathcal{G}_2 = (\{v_2\}, \emptyset, \emptyset)$, the *node augmentation* is an operation that creates a graph $\mathcal{G} = (\{v_1, v_2\}, \{e_{21}\}, \{d_{21}\})$ and is denoted by

$$\mathcal{G} := \mathcal{G}_1 \oplus \mathcal{G}_2 \qquad (8)$$

where $d_{21} = \|e_{21}\|$ and $\|e\|$ denotes the length of an edge $e$. By a slight abuse of notation, we write $\mathcal{G}_1 = \{v_1\}, \mathcal{G}_2 = \{v_2\}$ and then $\mathcal{G}$ can be rewritten as

$$\mathcal{G} := \{v_1\} \oplus \{v_2\}$$

Notice that a graph with two nodes and one edge is minimally rigid (due to Laman's theorem). Before we define the node augmentation for graphs with multiple nodes, we need to define the union of two graphs.

*Definition* 6. (union) Let $\mathcal{G}_1 = (\mathcal{V}_1, \mathcal{E}_1, \mathcal{W}_1)$ and $\mathcal{G}_2 = (\mathcal{V}_2, \mathcal{E}_2, \mathcal{W}_2)$ be two weighted graphs. The *union* of $\mathcal{G}_1$ and $\mathcal{G}_2$ is a graph

$$\mathcal{G} = \mathcal{G}_1 \cup \mathcal{G}_2 := (\mathcal{V}_1 \cup \mathcal{V}_2, \mathcal{E}_1 \cup \mathcal{E}_2, \mathcal{W}_1 \cup \mathcal{W}_2)$$
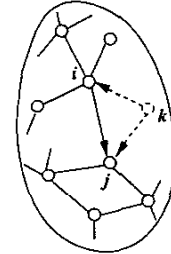


**Figure 4:** Augmentation of the node $k$ at the edge $(i, j)$ of a multi-node graph.

*Definition* 7. (node augmentation for multi-node graphs) Consider a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathcal{W})$ with $n \geq 2$ nodes and let $v_k \notin \mathcal{V}$ be the vertex of a single node graph $\mathcal{G}_0 = (\{v_k\}, \emptyset, \emptyset)$ with no edges. By a slight abuse of notation we denote $\mathcal{G}_0 = \{v_k\}$. Let $q_1, \ldots, q_n$ be a realization of $\mathcal{G}$ and assume $q_k$ does not coincide with any vertices of $\mathcal{G}$, i.e. $\|q_k - q_i\| > 0$, for $i = 1, \ldots, n$. The *node augmentation* at $(v_i, v_j)$ (i.e. an unordered pair of distinct vertices) is an operation on two graphs, respectively, with $n$ nodes and one node that creates a new graph $\mathcal{G}_a = (\mathcal{V}_a, \mathcal{E}_a, \mathcal{W}_a)$ with $n + 1$ nodes and $|\mathcal{E}_a| = |\mathcal{E}| + 2$ edges defined as the following

$$\begin{array}{l} \mathcal{V}_a = \mathcal{V} \cup \{v_k\}, \quad \mathcal{E}_a = \mathcal{E} \cup \{e_{ki}, e_{kj}\}, \\ \mathcal{W}_a = \mathcal{W} \cup \{\|e_{ki}\|, \|e_{kj}\|\}, \end{array} \qquad (9)$$

where $\|e\|$ denotes the length of the edge $e$. The node augmentation operation is denoted by

$$\mathcal{G}_a := \mathcal{G}|_{(v_i,v_j)} \oplus \{v_k\}, \qquad (10)$$

whenever an edge $e_{ij} \in \mathcal{E}$, this operation is equivalently expressed as

$$\mathcal{G}_a := \mathcal{G}|_{e_{ij}} \oplus \{v_k\}. \qquad (11)$$

The operation of node-augmentation is shown in Fig. 4. In this figure, the augmented node and edges are all drawn by dashed lines.

*Definition* 8. (a triangular graph) A *triangular graph* $T$ is a graph with three nodes, i. e. $\mathcal{V} = \{v_1, v_2, v_3\}$, and a set of three directed edges $\mathcal{E}_T \subset \mathcal{E} = \{e_{12}, e_{21}, e_{13}, e_{31}, e_{23}, e_{32}\}$.

A (special) triangular graph can be created using two consecutive node-augmentations as the following:

$$T = (\{v_1\} \oplus \{v_2\}) \oplus \{v_3\}$$

Since the graph $\{v_1\} \oplus \{v_2\}$ only has a single edge, the edge location in the last equation is dropped.

*Remark* 2. The operation of augmentation of a node $v_k$ at the pair $(v_i, v_j)$ can be equivalently represented by union of two graphs as follows. Let

$$T(v_k, v_i, v_j) := (\{v_k, v_i, v_j\}, \{e_{ki}, e_{kj}\}, \{\|e_{ki}\|, \|e_{kj}\|\}) \qquad (12)$$

denote a partial triangular graph with two edges and three nodes. Then $\mathcal{G}_a$, defined in (10), can be expressed as $\mathcal{G}_a = \mathcal{G} \cup T$.

Here is our first result on construction of rigid graphs via node augmentation.

**Theorem 2.** *Any rigid graph $\mathcal{G}$ in $\mathbb{R}^2$ remains rigid after node augmentation, i.e. node augmentation preserves rigidity in $\mathbb{R}^2$.*

**Proof.** See section A.1 in the Appendix of [12]. □

**Corollary 1.** *Any minimally rigid graph $\mathcal{G}$ in $\mathbb{R}^2$ remains minimally rigid after node augmentation.*

**Proof.** According to Theorem 2, the augmented graph $\mathcal{G}_a$ is rigid and has $|\mathcal{E}_{G_a}| = (2n - 3) + 2 = 2(n+1) - 3$ edges. Thus, $\mathcal{G}_a$ satisfies the relation $|\mathcal{E}_{G_a}| = 2|\mathcal{V}_{G_a}| - 3$. The fact that $G_a$ is an essential subgraph of itself follows from the proof of Theorem 2 and that establishes every Laman subgraph $Y_a$ of $\mathcal{G}_a$ is proper. □

**Corollary 2.** *The augmented edges in node augmentation of a minimally rigid graph are independent edges of the obtained graph.*

**Proof.** This follows from the fact that the obtained graph after node augmentation is minimally rigid and all of its edges are independent.

The following corollary provides a simple proof of rigidity for minimally rigid graphs.

**Corollary 3.** *Any graph that can be constructed by an ordered sequence of node augmentations starting from a single node is minimally rigid.*

*Example* 1. Here is a sequence of nodes and edges that create a (minimally) rigid graph shown in Fig. 5 (a), (b).

$$
\begin{aligned}
\mathcal{G}_a : \quad & 1; 2, (2,1); 3, (3,1), (3,2); 4, (4,2), (4,3); 5, (5,2) \\
& , (5,4); 6, (6,4), (6,3). \\
\mathcal{G}_b : \quad & 1; 2, (2,1); 3, (3,1), (3,2); 4, (4,2), (4,3); 5, (5,3) \\
& , (5,4); 6, (6,4), (6,5); 7, (7,5), (7,6). \\
\mathcal{G}_c : \quad & 1; 2, (2,1); 3, (3,1), (3,2); 4, (4,2), (4,3). \\
\mathcal{G}_d : \quad & 1; 2, (2,1); 3, (3,1), (3,2); 4, (4,2), (4,3); 5, (5,3) \\
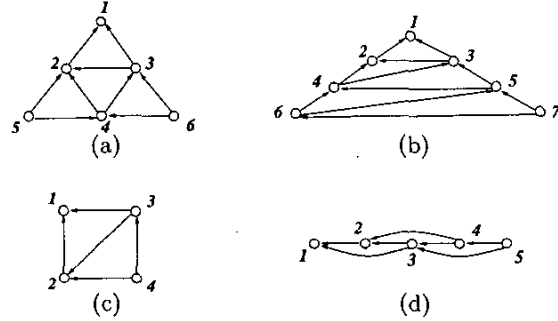& , (5,4).
\end{aligned}
$$

$$(13)$$



**Figure 5:** (a) A rigid graph $\mathcal{G}_a$ with $n = 6$ nodes and $n_e = 9$ edges, (b) A rigid graph $\mathcal{G}_b$ representing a $V$-formation of $n = 7$ vehicles with $n_e = 11$ edges, (c) a square graph $\mathcal{G}_c$ with a diagonal edge representing a diamond formation of four vehicles, and (d) A rigid graph $\mathcal{G}_d$ with $n = 5$ nodes on a line representing a platoon of five vehicles.

## 3.2 Rejoining Two Graphs

As described earlier in the introduction, the *rejoin operation* of two subgroups of vehicles is very important after they pass an obstacle (see Figures 1 (a), (b)). In this section, we present the graph theoretical tools/operations that create a rigid composite graph by combining the graphs associated with each subgroup of agents/vehicles.

Let $\mathcal{G}_1 = (\mathcal{V}_1, \mathcal{E}_1, \mathcal{W}_1)$ and $\mathcal{G}_2 = (\mathcal{V}_2, \mathcal{E}_2, \mathcal{W}_2)$ be two graphs with disjoint set of vertices $\mathcal{V}_1 \cap \mathcal{V}_2 = \emptyset$ and $n_1 \geq 2$ and $n_2 \geq 2$ nodes, respectively. We say $\mathcal{G}_1$ and $\mathcal{G}_2$ are *edge-attachable* at $e_{ij} = (v_i, v_j) \in \mathcal{E}_1$ and $e_{kl} = (v_k, v_l) \in \mathcal{E}_2$ iff $e_{ij}$ and $e_{kl}$ have the equal lengths, i.e. $\|e_{ij}\| = \|e_{kl}\|$. Before describing the operation of

edge-attachment of two graphs, we need to introduce three basic operations on graphs. Namely, the operations of renaming of vertices, edge-addition, and edge-subtraction in a graph.

*Definition 9.* (single-node renaming) Let $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ be a graph and $v_k \notin \mathcal{V}$ be an extra node. The operation of *renaming* $v_i \in \mathcal{V}$ in $\mathcal{G}$ by $v_k$ means creating a new graph $\mathcal{G}' = (\mathcal{V}', \mathcal{E}')$ such that $\mathcal{V}' = (\mathcal{V} \setminus \{v_i\}) \cup \{v_k\}$ and $\mathcal{E}' = (\mathcal{E} \setminus \mathcal{E}_i) \cup \mathcal{E}_k$ where

$$
\begin{aligned}
\mathcal{E}_i &= \{e \in \mathcal{E} : e = (v_i, v_j) \vee e = (v_j, v_i)\}, \\
\mathcal{E}_k &= \{(v_i, v_k) : (v_i, v_j) \in \mathcal{E}\} \cup \{(v_k, v_i) : (v_j, v_i) \in \mathcal{E}\}
\end{aligned}
\tag{14}
$$

*Definition 10.* (multi-node renaming) Similarly, more than one node in a graph can be renamed. The operation of renaming an ordered list of nodes $v_1, v_2, \ldots, v_m$ in $\mathcal{G}$ by $v'_1, v'_2, \ldots, v'_m$ is called *multi-node renaming* and is denoted by

$$
\mathcal{G}' = \mathrm{Ren}(\mathcal{G}; v_1, v_2, \ldots, v_m | v'_1, v'_2, \ldots, v'_m)
$$

This operation means creating a new graph $\tilde{\mathcal{G}}$ iteratively as the following

$$
\begin{aligned}
\mathcal{G}^0 &:= \mathcal{G} \\
\mathcal{G}^k &:= \mathrm{Ren}(\mathcal{G}^{k-1}; v_k | v'_k), k = 1, \ldots, m \\
\mathcal{G}' &:= \mathcal{G}^m
\end{aligned}
\tag{15}
$$

With a minor abuse of notation, the renaming operation can be applied to the set of edges of a graph instead of the whole graph.

*Definition 11.* (edge-addition/subtraction) Let $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ be a graph and let $\mathcal{G}^+ := (\mathcal{V}, \mathcal{E} \cup \{e\})$. Then the operation $[\cdot]_e^+$ on a graph defined as $\mathcal{G}^+ = [\mathcal{G}]_e^+$ is called *edge-addition*. Similarly, let $\mathcal{G}^- := (\mathcal{V}, \mathcal{E} \setminus \{e\})$. Then, the operation $[\cdot]_e^-$ on a graph defined by $\mathcal{G}^- = [\mathcal{G}]_e^-$ is referred to as *edge-subtraction*. Notice that for weighted graphs, edge addition (or subtraction) operation adds (or subtracts) the element $w = \|e\|$ to (from) the set of weights $\mathcal{W}$.
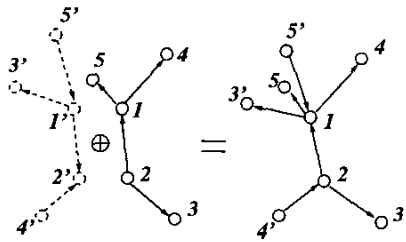


**Figure 6:** Edge-attachment between two graphs at the edges $(1, 2)$ and $(1',2')$ with equal length.

*Definition 12.* (edge-attachment) Let $\mathcal{G}_1 = (\mathcal{V}_1, \mathcal{E}_1, \mathcal{W}_1)$ and $\mathcal{G}_2 = (\mathcal{V}_2, \mathcal{E}_2, \mathcal{W}_2)$ be two graphs that are edge-attachable at $e_{ij} \in \mathcal{E}_1$ and $e_{kl} \in \mathcal{E}_2$. Define

$$
\mathcal{G}_2^- = [\mathcal{G}_2]_{e_{kl}}^- := (\mathcal{V}_2, \mathcal{E}_2 \setminus \{e_{kl}\}, \mathcal{W}_2 \setminus \{w_{kl}\}),
$$

and let $\mathcal{G}_2' = \mathrm{Ren}(\mathcal{G}_2^-; v_k, v_l | v_i, v_j)$. Then, the *edge-attachment operation* of $\mathcal{G}_2 = (\mathcal{V}_2, \mathcal{E}_2)$ to $\mathcal{G}_1 = (\mathcal{V}_1, \mathcal{E}_1)$ at $e_{ij} \in \mathcal{E}_1$ is defined as

$$
\mathcal{G} := \mathcal{G}_1 \cup \mathcal{G}_2'
\tag{16}
$$

and denoted by

$$
\mathcal{G} = \mathcal{G}_1|_{e_{ij}} \oplus \mathcal{G}_2|_{e_{kl}}
\tag{17}
$$

We call $\mathcal{G}$ the *attachment* (graph) of $\mathcal{G}_1$ and $\mathcal{G}_2$. The overall edge-attachment operation can be equivalently expressed as follows

$$
\mathcal{G}_1|_{e_{ij}} \oplus \mathcal{G}_2|_{e_{kl}} := \mathcal{G}_1 \cup \mathrm{Ren}([\mathcal{G}_2]_{e_{kl}}^-; v_k, v_l | v_i, v_j)
\tag{18}
$$

Notice that the edge-attachment operation that is shown schematically in Fig. 6 creates a new graph $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathcal{W})$ with $\mathcal{V} = \mathcal{V}_1 \cup (\mathcal{V}_2 \setminus \{v_k, v_l\})$, $\mathcal{E} = \mathcal{E}_1 \cup \mathrm{Ren}(\mathcal{E}_2 \setminus \{e_{kl}\}; v_k, v_l | v_i, v_j)$, and $\mathcal{W} = \|\mathcal{E}\|$. Furthermore, the obtained graph $\mathcal{G}$ has $n = (n_1 + n_2) - 2$ nodes ($n \geq 2$) and $|\mathcal{E}| = |\mathcal{E}_1| + |\mathcal{E}_2| - 1$ edges.

*Example 2.* Consider two triangular graphs $T_1 = (\{v_1, v_2, v_3\}, \{e_{12}, e_{23}, e_{31}\})$ and $T_2 = (\{v_4, v_5, v_6\}, \{e_{45}, e_{56}, e_{64}\})$ and let $\|e_{23}\| = \|e_{56}\|$ then the graph $\mathcal{G} = T_1|_{e_{23}} \oplus T_2|_{e_{56}}$ is a graph with the set of vertices $\mathcal{V} = \{v_1, v_2, v_3, v_4\}$ and the set of edges $\mathcal{E} = \{e_{12}, e_{23}, e_{31}, e_{42}, e_{34}\}$.

Our main motivation to define the operation of edge-attachment of two weighted graphs is the following result.

**Theorem 3.** *(edge-attachment) Let $\mathcal{G}_1$ and $\mathcal{G}_2$ be two rigid graphs with disjoint set of vertices. Assume that the corresponding essential subgraphs $\mathcal{H}_1$ and $\mathcal{H}_2$ of these graphs possess attachable edges $e_{ij} \in \mathcal{E}_{\mathcal{H}_1}$ and $e_{kl} \in \mathcal{E}_{\mathcal{H}_2}$. Then the attachment $\mathcal{G} = \mathcal{G}_1|_{e_{ij}} \oplus \mathcal{G}_2|_{e_{kl}}$ is a rigid graph.*
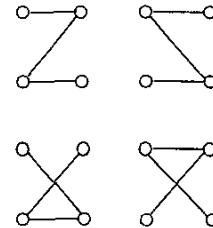
**Proof.** See section A.2 in the Appendix of [12]. □



**Figure 7:** A $Z$-link in four possible configuration of edges.

*Definition 13.* ($Z$-link) We refer to a *bipartite graph* $\mathcal{K}_{2,2}$ with three edges, shown in Fig. 7, as a $Z$-*link*.

*Remark 3.* The name "$Z$-link" comes from the shape of the graph in the upper left corner of Fig. 7.

*Definition* 14. (rejoining two graphs) The operation of *rejoining two graphs* $\mathcal{G}_1 = (\mathcal{V}_1, \mathcal{E}_1)$ and $\mathcal{G}_2 = (\mathcal{V}_2, \mathcal{E}_2)$ via a *Z-link*, $Z = (\mathcal{V}_Z, \mathcal{E}_Z)$, means creating a new graph

$$\mathcal{G} = \mathcal{G}_1|_{e_1} \oplus_Z \mathcal{G}_2|_{e_2} \qquad (19)$$

where $\mathcal{G} = \mathcal{G}_1 \cup Z \cup \mathcal{G}_2$ and $\mathcal{V}_Z = \mathcal{V}(e_1) \cup \mathcal{V}(e_2)$, i.e. the vertices of the $Z$-link are the end-points of two edges $e_1$ and $e_2$. The joining operation is schematically shown in Fig. 8. The edges that are removed after edge-attachment are shown with dashed lines.
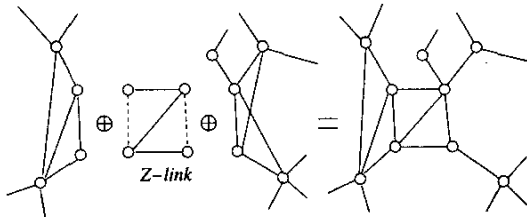


**Figure 8:** Joining two graphs using a $Z$-link (here the direction of the edges can be chosen arbitrarily).

Here is an important result that states how two graphs need to be connected such that the obtained graph after connection is rigid. This is particularly useful in rejoin maneuvers of two formations of multiple vehicles.

**Theorem 4.** *Let $\mathcal{G}_1, \mathcal{G}_2$ be two rigid graph each with more than two nodes. Assume $e_1, e_2$ are edges of essential subgraphs of $\mathcal{G}_1, \mathcal{G}_2$, respectively. Then, the connection of $\mathcal{G}_1$ and $\mathcal{G}_2$ via a $Z$-link given by $\mathcal{G} = \mathcal{G}_1|_{e_1} \oplus_Z \mathcal{G}_2|_{e_2}$ is a rigid graph.*

**Proof.** See section A.3 in the Appendix of [12]. □

**Corollary 4.** *Two minimally rigid graphs that are connected using a $Z$-link construct a minimally rigid graph.*

*Example* 3. In Fig. 2, two triangular formations of vehicles are rejoined using a $Z$-link to create a formation of six vehicles with a minimally rigid interconnection graph.

*Remark* 4. *Splitting* a group of vehicles into two disjoint subgroups is discussed in [12].

## 4 Conclusion

In this paper, we provide a unified graph-theoretical framework that allows us to formally define formations of multiple vehicles and their stabilization issues. We clarified the important role of graph rigidity and minimally rigid graphs in construction of structural potential functions and manipulation of multiple formations. This includes formal representation of split, rejoin, and reconfiguration maneuvers for formations of multi-vehicle systems. We presented a hybrid system framework for consecutive execution of a set of

maneuvers for a group of agents/vehicles that allows high-level planning of operations in multi-vehicle systems. We introduce an algebra that formalizes performing some basic operations on graphs and allows creation of larger rigid-by-construction graphs by combining smaller rigid subgraphs.

### References

[1]   N. E. Leonard and E. Fiorelli, "Virtual leaders, artificial potentials, and coordinated control of groups," *Proc. of the 40th IEEE Conference on Decision and Control*, Orlando, FL, Dec. 2001.

[2]   R. Olfati-Saber and R. M. Murray, "Distributed cooperative control of multiple vehicle formations using structural potential functions," *The 15th IFAC World Congress*, June 2002, http://www.cds.caltech.edu/~murray/papers/ 20011_om02-ifac.html.

[3]   M. Mesbahi and F. Y. Hadegh, "Formation flying of multiple spacecraft via graphs, matrix inequalities, and switching," *AIAA Journal of Guidance, Control, and Dynamics*, vol. 24, no. 2, pp. 369–377, March 2000.

[4]   A. Fax and R. M. Murray, "Information Flow and Cooperative Control of Vehicle Formations," *The 15th IFAC World Congress*, June 2002.

[5]   T. Eren, P. N. Belhumeur, B. D. O. Anderson, and S. A. Morse, "A framework for maintaing formations based on rigidity," *Proceedings of the 2002 IFAC World Congress*, July 2002.

[6]   J. Graver and H. Servatius, B. Servatius, *Combinatorial Rigidity*, vol. 2 of *Graduate Studies in Mathematics*, American Mathematical Society, 1993.

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

[8]   L. Lovász and Y. Yemini, "On generic rigidity in the plane," *SIAM J. Alg. Disc. Meth.*, vol. 3, no. 1, pp. 91–98, March 1982.

[9]   B. Roth, "Rigidity and flexible frameworks," *The American Mathematical Monthly*, vol. 88, pp. 6–21, January 1982.

[10]   H. Gluck, "Almost all simply connected closed surfaces are rigid," in *Geometric Topology*, number 438 in Lecture Notes In Mathematics, pp. 225–239. Springer-Verlag, Berlin, 1975.

[11]   H. N. Gabow and H. H. Westermann, "Forests, frames, and games: algorithms for matroid sums and applications," *Proc. 20th Ann. ACM Symp. on the Theory of Computing*, pp. 407–421, 1988.

[12]   R. Olfati-Saber and R. M. Murray, "Graph rigidity and distributed formation stabilization of multi-vehicle systems," Tech. Rep. CIT-CDS 2002-001, California Institute of Technology, Control and Dynamical Systems, Pasadena, California, February 2002.