

# Fast Multiagent Formation Stabilization with Sparse Universally Rigid Frameworks

Zhonggang Li, Geert Leus, and Raj Thilak Rajan

*Signal Processing Systems, Department of Microelectronics, Faculty of EEMCS*

*Delft University of Technology*

Delft, The Netherlands

{z.li-22, g.j.t.leus, r.t.rajan}@tudelft.nl

**Abstract**—Affine formation control (AFC) is a distributed networked control system that has recently received increasing attention in various applications. AFC is typically achieved using a generalized consensus system where the stress matrix, which encodes the graph structure, is used instead of a graph Laplacian. Universally rigid frameworks (URFs) guarantee the existence of the stress matrix and have thus become the guideline for such a network design. In this work, we propose a convex optimization framework to design the stress matrix for AFC without predefining a rigid graph. We aim to find a resulting network with a reduced number of communication links, but still with a fast convergence speed. We show through simulations that our proposed solutions can yield a more sparse graph, while admitting a faster convergence compared to the state-of-the-art solutions.

**Index Terms**—network design, consensus, graph rigidity, formation control, multiagent systems, stress matrix

## I. INTRODUCTION

Distributed formation control is an essential task for robotic swarming applications [1], [2], [3], where agents use relative information such as interagent distances, relative positions, etc., to achieve and maintain a desired geometric pattern in two- or three-dimensional space. Such a system is typically characterized by a framework [4] consisting of (a) a configuration, which represents the collection of agent positions forming a geometric pattern, and (b) a graph where the edges denote the communication links for information exchange. Such a networked system shares similarities with sensor networks, relative localization, distributed optimization, etc., where the communication pattern and capacity play a critical role [5], [6], [7].

Recently, affine formation control (AFC) has gained increasing attention due to its flexibility in maneuvering while maintaining coordinated motion [8]. Unlike traditional formation control that enforces rigid geometric constraints, AFC allows for controlled scaling, shearing, etc., making it particularly useful for dynamic environments and adaptive mission planning. AFC follows a consensus-like framework, where control inputs are derived as linear combinations of position differences between neighboring agents. As compared to the

conventional average consensus using the graph Laplacian, AFC adopts a stress matrix - a generalization of the Laplacian with weighted edges. The universal rigidity of the associated framework guarantees the existence of stress with specific properties that enable AFC. Universally rigid frameworks (URFs) are also applied in network localization [9], tensegrity frameworks [10], etc.

Interagent communication is essential in distributed formation control, making the reduction of communication load a crucial necessity, due to bandwidth and power constraints of agents. However, a sparse network with few edges is typically less efficient in information exchange, and thus slows down the convergence of consensus-based algorithms including AFC. This has been an ongoing discussion in consensus theory involving the algebraic connectivity [11], [12]. In AFC, designing the stress matrix with fewer edges while preserving rigidity and convergence speed is challenging. Earlier mathematical literature discusses URF design using special geometries [13], [10], later followed by numerical stress calculations for AFC applications [14], [8]. However, manual graph design becomes impractical for larger networks, and structured geometries like Grünbaum polygons [13] can create unbalanced networks that hinder robustness. Thus, generative methods [15], [16] that directly design a possibly sparse stress matrix from a given configuration without predefined graphs are preferred. A sparse nullspace reconstruction approach [15] achieves substantial sparsification but fails to maintain convergence from an arbitrary initial configuration in many cases. More recently, mixed-integer semidefinite programming (MISDP) is adopted in [16] for sparse stress design with optimized convergence speed, but raises concerns about optimality and computational complexity.

In this paper, we present an efficient convex optimization framework to design the stress matrix with two key objectives, (a) network sparsification and (b) convergence acceleration, and we further give insights on how to balance these two rather conflicting objectives. In Section II, we introduce the preliminaries of graph rigidity and how stress matrices play a critical role in AFC. We provide a preliminary problem formulation based on the properties of stress and our optimization objective, which turns out to be a non-convex problem. In Section III, we relax the non-convex problem into a concise and tractable convex problem with insights into the choice of

This work is partially funded by the Sensor AI Lab, under the AI Labs program of Delft University of Technology, and by the RVO (Missie gedreven Onderzoek, Ontwikkeling en Innovatie (MOOI)) Aerial Quadcopter Units for Aquatic Flow Investigation and Nautical Data (AQUAFIND) project.

hyperparameters. Finally, we compare our proposed solution with the aforementioned state-of-art using several numerical examples in Section IV, before reaching a conclusion and giving potential future research directions in Section V.

*Notations.* Vectors and matrices are represented by lowercase and uppercase boldface letters, respectively such as  $\mathbf{a}$  and  $\mathbf{A}$ . Sets and graphs are represented using calligraphic letters, e.g.,  $\mathcal{A}$ . Vectors of length  $N$  of all ones and zeros are denoted by  $\mathbf{1}_N$  and  $\mathbf{0}_N$ , respectively, and their matrix versions are similarly  $\mathbf{1}_{M \times N}$  and  $\mathbf{0}_{M \times N}$ . An identity matrix of size  $N$  is denoted by  $\mathbf{I}_N$ . The Kronecker product is denoted by  $\otimes$ , which has relationships with the vectorization operator  $\text{vec}(\cdot)$  [17]. The  $\text{diag}(\cdot)$  operator creates a diagonal matrix from a vector and  $\text{tr}(\cdot)$  denotes the trace operator.  $\lambda_k(\mathbf{A})$  denotes the  $k$ -th smallest eigenvalue of a symmetric matrix  $\mathbf{A}$ .

## II. FUNDAMENTALS

### A. Graphs and Rigidity Theory

Consider  $N$  mobile agents moving in  $D$ -dimensional Euclidean space where  $N \geq D + 1$ . An undirected graph  $\mathcal{G} = (\mathcal{V}, \mathcal{E})$  is used to model a communication network, where the vertices  $\mathcal{V} = \{1, \dots, N\}$  denote the agents, and the edges  $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$  denote the pairwise interactions, e.g., information exchange. We use  $N = |\mathcal{V}|$  and  $M = |\mathcal{E}|$  as a shorthand notation for the number of vertices and edges, respectively. The set of neighbors of a node  $i$  is defined as  $\mathcal{N}_i = \{j \in \mathcal{V} : (i, j) \in \mathcal{E}\}$ . Let  $\mathbf{p}_i \in \mathbb{R}^D$  be the position of node  $i \in \mathcal{V}$  and the collection of all nodes, the *configuration*, is  $\mathbf{P} = [\mathbf{p}_1, \dots, \mathbf{p}_N] \in \mathbb{R}^{D \times N}$ . A *generic configuration* [18] has algebraically independent node coordinates, i.e., no geometric constraints among nodes. Classic nongeneric configurations include nodes forming a line, a regular polygon, etc.

A *framework*  $\mathcal{F} = (\mathcal{G}, \mathbf{P})$  is a tuple of the graph and its associated configuration. Intuitively, the rigidity of frameworks can be judged by whether a different configuration exists given the distances between the nodes on the edges of the graph. More formally, from [14], let  $\mathbf{P}' = [\mathbf{p}'_1, \dots, \mathbf{p}'_N] \in \mathbb{R}^{D \times N}$ , then two frameworks  $(\mathcal{G}, \mathbf{P})$  and  $(\mathcal{G}, \mathbf{P}')$  are equivalent if  $\forall (i, j) \in \mathcal{E}, \|\mathbf{p}_i - \mathbf{p}_j\|_2 = \|\mathbf{p}'_i - \mathbf{p}'_j\|_2$ , and they are congruent if  $\forall i, j \in \mathcal{V}, \|\mathbf{p}_i - \mathbf{p}_j\|_2 = \|\mathbf{p}'_i - \mathbf{p}'_j\|_2$ . The global rigidity of a framework  $\mathcal{F}$  defined in  $\mathbb{R}^D$  requires that all frameworks that are equivalent to  $\mathcal{F}$  are also congruent to it. Note that global rigidity is sometimes simply referred to as rigidity. Frameworks that are not globally rigid are *flexible*. The universal rigidity of a framework  $\mathcal{F}$  defined in  $\mathbb{R}^D$  requires  $\mathcal{F}$  to be globally rigid in  $\mathbb{R}^{D'}$  for any  $D' \geq D$ . Fig. 1 presents a few examples to illustrate the rigidities.

Universal rigidity is algebraically represented by an *equilibrium stress*  $\omega_{ij} \in \mathbb{R}$  for every edge  $(i, j) \in \mathcal{E}$ , leading to a set of weights associated with the edges, which satisfy

$$\sum_{(i,j) \in \mathcal{E}} \omega_{ij} (\mathbf{p}_i - \mathbf{p}_j) = \mathbf{0}_D. \quad (1)$$

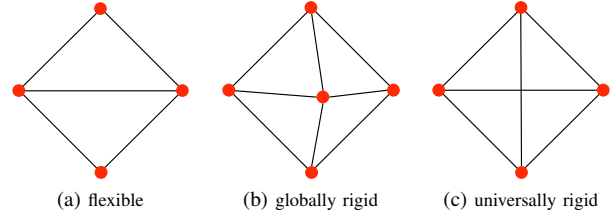


Fig. 1. Examples of frameworks in  $\mathbb{R}^2$  with increasing rigidity [14]

A more compact form of (1) is  $\mathbf{\Omega P}^\top = \mathbf{0}_{N \times D}$ , where  $\mathbf{\Omega} \in \mathbb{R}^{N \times N}$  is called the *stress matrix* and is defined as

$$[\mathbf{\Omega}]_{ij} = \begin{cases} 0, & \text{if } (i, j) \notin \mathcal{E} \\ -\omega_{ij}, & \text{if } i \neq j, (i, j) \in \mathcal{E} \\ \sum_{k \in \mathcal{N}_i} \omega_{ik} & \text{if } i = j \end{cases} \quad (2)$$

Alternatively, the stress matrix can be defined using the graph incidence matrix  $\mathbf{B} \in \mathbb{R}^{N \times M}$  as

$$\mathbf{\Omega} = \mathbf{B} \text{diag}(\boldsymbol{\omega}) \mathbf{B}^\top, \quad (3)$$

where  $\boldsymbol{\omega} \in \mathbb{R}^M$  is a vector containing all the equilibrium stresses. Note that  $\mathbf{\Omega}$  reduces to a standard graph Laplacian if  $\text{diag}(\boldsymbol{\omega}) = \mathbf{I}_M$ , i.e., equal weights for the edges. It also has  $\mathbf{1}_N$  in the nullspace like the Laplacian. The following theorem establishes the important properties of  $\mathbf{\Omega}$  related to universal rigidity.

**Theorem 1.** (*Universally Rigid Frameworks and Stress Matrices*) Given a framework  $\mathcal{F} = (\mathcal{G}, \mathbf{P})$  with  $\mathbf{P}$  being a generic configuration,  $\mathcal{F}$  is universally rigid if and only if there exists a positive semidefinite stress matrix  $\mathbf{\Omega}$  with rank  $N - D - 1$ .

**Proof.** See [19], [8], [16]. ■

Although a generic configuration is generally assumed for mathematical guarantees, nongeneric geometries are more interesting for formation control applications. It is worth mentioning that our stress design approach works for both generic and nongeneric configurations.

### B. Consensus-Based Formation Control

Let  $\mathbf{z} = [\mathbf{z}_1^\top, \dots, \mathbf{z}_N^\top]^\top \in \mathbb{R}^{DN}$ , where  $\mathbf{z}_i \in \mathbb{R}^D, \forall i \in \mathcal{V}$ , be the true positions of agents that are governed by the single-integrator dynamics  $\dot{\mathbf{z}}_i = \mathbf{u}_i$  where  $\mathbf{u}_i$  is a velocity control input to be computed by agent  $i$ . Furthermore, recall the stress matrix from (1) that satisfies  $\mathbf{\Omega P}^\top = \mathbf{0}$ , which can also be written as

$$(\mathbf{\Omega} \otimes \mathbf{I}_D) \mathbf{p} = \mathbf{0}, \quad (4)$$

where  $\mathbf{p} = \text{vec}(\mathbf{P})$  using the Kronecker-vectorization property. As such, a consensus-like control law [14], [8] can be designed as  $\mathbf{u}_i = -\sum_{j \in \mathcal{N}_i} \omega_{ij} (\mathbf{z}_i - \mathbf{z}_j), \forall i \in \mathcal{V}$ , such that the collective dynamics lead to a linear time-invariant system

$$\dot{\mathbf{z}} = -(\mathbf{\Omega} \otimes \mathbf{I}_D) \mathbf{z}. \quad (5)$$

It is straightforward to see from (4) that  $\mathbf{p}$  is an equilibrium of system (5), which is allowed by  $\mathbf{\Omega}$  that has a  $(D + 1)$ -dimensional nullspace. This is advantageous over

the conventional average consensus system, where the graph Laplacian is used because the graph Laplacian has only a 1-dimensional nullspace corresponding to the  $\mathbf{1}$  vector, i.e., only configurations with all nodes in the same location can be in equilibrium.

Moreover, since  $\Omega$  is positive semidefinite, system (5) is globally and exponentially stable. We then claim that a formation defined by the framework  $\mathcal{F} = (\mathcal{G}, \mathbf{P})$  is stabilizable using system (5), i.e., the formation asymptotically converges to the solution space containing the equilibrium  $\mathbf{p}$  given any initialization. Note that the targeted equilibrium  $\mathbf{p}$  can be reached up to an affine transformation (hence its usefulness for AFC), but can also be uniquely determined given a few anchor nodes (which are commonly named leaders, e.g., in [8]).

### C. Problem Formulation

As discussed, the core element of the formation control system (5) is the stress matrix with the following properties:

- 1)  $\Omega$  is positive semidefinite (PSD);
- 2)  $\Omega$  has rank  $N - D - 1$ ;
- 3)  $\Omega$  has rows of configuration  $\mathbf{P}$  and  $\mathbf{1}_N$  in the nullspace.

Given a configuration  $\mathbf{P}$ , we seek a sparse stress matrix satisfying the above properties representing a loosely connected graph, while maximizing the system's convergence rate. In consensus theory, the second smallest eigenvalue of the Laplacian governs convergence speed [12], as it determines the slowest decaying component. Prior work has aimed to maximize this eigenvalue for faster convergence [11]. Similarly, for stress matrices, we aim to maximize the smallest nonzero eigenvalue, i.e., the  $(D+2)$ -th smallest eigenvalue of  $\Omega$  or  $\lambda_{D+2}(\Omega)$ . We show later via numerical examples that this goal can be traded off for graph sparsification.

## III. NETWORK TOPOLOGY SPARSIFICATION

In this section, we first propose a preliminary formulation directly related to the objectives and the constraints, which turns out to be a non-convex problem. We then focus on convexifying the formulation into a tractable convex optimization and give insights into the formulation. Recall from (3) that the stress matrix can be constructed using the graph incidence matrix  $\mathbf{B}$  given a stress vector  $\bar{\omega}$ . We initialize the graph as a complete graph with an incidence matrix  $\bar{\mathbf{B}} \in \mathbb{R}^{N \times \bar{M}}$  where there are  $\bar{M} = \frac{N(N-1)}{2}$  edges. We then aim to find a sparse vector  $\bar{\omega} \in \mathbb{R}^{\bar{M}}$  such that  $\Omega$  is sparse using (3), but using the notation with bars. The effective number of edges from the sparse vector  $\Omega$  will be  $M = \|\bar{\omega}\|_0$ . Hence we pose the following problem  $\mathcal{P}_0$ :

$$\mathcal{P}_0 : \quad \underset{\bar{\omega}, \Omega}{\text{minimize}} \quad \|\bar{\omega}\|_0 - \alpha \lambda_{D+2}(\Omega) \quad (6a)$$

$$\text{subject to} \quad \Omega = \bar{\mathbf{B}} \text{diag}(\bar{\omega}) \bar{\mathbf{B}}^\top \quad (6b)$$

$$\text{rank}(\Omega) = N - D - 1 \quad (6c)$$

$$\Omega \succeq 0 \quad (6d)$$

$$\Omega \bar{\mathbf{P}}^\top = \mathbf{0} \quad (6e)$$

where  $\alpha$  is a weighting parameter,  $\lambda_{D+2}(\Omega)$  is the smallest nonzero eigenvalue of  $\Omega$ , and  $\bar{\mathbf{P}} = [\mathbf{P}^\top, \mathbf{1}_N]^\top \in \mathbb{R}^{(D+1) \times N}$ . As can be observed,  $\mathcal{P}_0$  is difficult to solve due to the L0-norm and the rank constraint. In the next section, we convexify this problem to a tractable semidefinite program (SDP).

### A. Convexifying the Constraints

Assuming a full-rank  $\bar{\mathbf{P}}$ , we let  $\mathbf{Q} \in \mathbb{R}^{N \times (N-D-1)}$  span the kernel of  $\bar{\mathbf{P}}$  with orthonormal columns. Then, (6c), (6d), and (6e) imply that  $\mathbf{Q}^\top \Omega \mathbf{Q} \in \mathbb{R}^{(N-D-1) \times (N-D-1)}$  is a positive definite (PD) matrix, since the null-subspace of  $\Omega$  is projected out by  $\mathbf{Q}$ , i.e.,  $\mathbf{Q}^\top \Omega \mathbf{Q} \succ 0$ . Further substituting (3), we have  $\Psi \text{diag}(\bar{\omega}) \Psi^\top \succ 0$ , where  $\Psi = \mathbf{Q}^\top \bar{\mathbf{B}}$ .

Substituting (3) into (6e), we have  $\bar{\mathbf{P}} \bar{\mathbf{B}} \text{diag}(\bar{\omega}) \bar{\mathbf{B}}^\top = \bar{\mathbf{P}} \bar{\mathbf{B}} \text{diag}(\bar{\omega}) [\bar{\mathbf{b}}_1, \dots, \bar{\mathbf{b}}_N] = \mathbf{0}$  where  $\bar{\mathbf{b}}_i \in \mathbb{R}^{\bar{M}}$ ,  $\forall i \in \mathcal{V}$  is the  $i$ -th column of  $\bar{\mathbf{B}}^\top$ . Observe that  $\bar{\mathbf{P}} \bar{\mathbf{B}} \text{diag}(\bar{\omega}) \bar{\mathbf{b}}_i = \bar{\mathbf{P}} \bar{\mathbf{B}} \text{diag}(\bar{\mathbf{b}}_i) \bar{\omega} = \mathbf{0}$ . As such, we can construct a matrix  $\mathbf{E} \in \mathbb{R}^{N(D+1) \times \bar{M}}$  with the structure

$$\mathbf{E} = \begin{bmatrix} \bar{\mathbf{P}} \bar{\mathbf{B}} \text{diag}(\bar{\mathbf{b}}_1) \\ \vdots \\ \bar{\mathbf{P}} \bar{\mathbf{B}} \text{diag}(\bar{\mathbf{b}}_N) \end{bmatrix}, \quad (7)$$

such that  $\mathbf{E} \bar{\omega} = \mathbf{0}$ , which can replace constraint (6e).

### B. Eigenvalue Maximization

We now give an explicit expression for the maximization of  $\lambda_{D+2}(\Omega)$ . Denoting  $\psi = \text{diag}(\Psi^\top \Psi)$ , we show that maximizing  $\psi^\top \bar{\omega}$  with a bounded  $\|\Omega\|_2$  is equivalent to the eigenvalue maximization problem. To start, recollecting that  $\mathbf{Q}^\top \Omega \mathbf{Q} = \Psi \text{diag}(\bar{\omega}) \Psi^\top$ , we obtain

$$\begin{aligned} \text{tr}(\mathbf{Q}^\top \Omega \mathbf{Q}) &= \text{tr}(\Psi \text{diag}(\bar{\omega}) \Psi^\top) = \text{tr}(\Psi^\top \Psi \text{diag}(\bar{\omega})) \\ &= \text{tr}(\text{diag}(\Psi^\top \Psi) \bar{\omega}) = \psi^\top \bar{\omega}. \end{aligned} \quad (8)$$

Hence, the sum of the eigenvalues of the PD matrix  $\mathbf{Q}^\top \Omega \mathbf{Q}$ , or equivalently the PSD matrix  $\Omega$  (since  $\mathbf{Q}$  has orthonormal columns), can be represented by  $\psi^\top \bar{\omega}$  where  $\psi$  is known. Then if we use an additional constraint  $\|\Omega\|_2 \leq \beta$  where we limit the largest eigenvalue of  $\Omega$  to  $\beta > 0$ ,  $\lambda_{D+2}(\Omega)$  is maximized when  $\lambda_{D+2}(\Omega) = \dots = \lambda_{\max}(\Omega) = \beta$  and the condition number  $\kappa(\Omega)$  is minimized to 1 at the same time. This eigenvalue maximization is guaranteed when there are no other objectives and constraints and will play as a relaxation under our proposed constrained formulation. It is worth mentioning that trace regularization is also commonly seen in network design literature such as [11]. Additionally, although we do not emphasize the minimization of the condition number in this work, it is shown in [16] that such minimization can improve the robustness of AFC against time delays.

### C. Proposed Framework

A common convex relaxation is that of the L0-norm to an L1-norm for sparsity, i.e.,  $\|\bar{\omega}\|_0$  to  $\|\bar{\omega}\|_1$  in the objective.

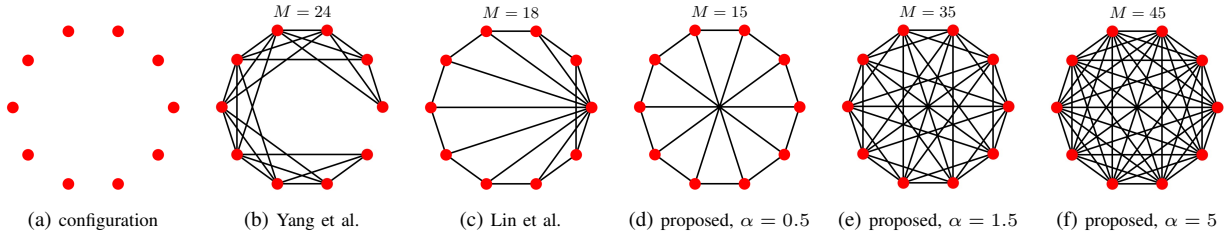


Fig. 2. Resulting graphs of our proposed solutions compared with the state-of-the-art. A 10-node regular polygon has a maximum of  $\bar{M} = 45$  edges. The recommended value of  $\alpha$  for a sparse solution is  $1/\|\psi\|_\infty = 0.51$ . The structure from [13] is used for Lin et al.

Finally, by combining all the convexified constraints and the objectives above, we present our proposed formulation  $\mathcal{P}_1$

$$\mathcal{P}_1 : \quad \underset{\bar{\omega}}{\text{minimize}} \quad \|\bar{\omega}\|_1 - \alpha \psi^\top \bar{\omega} \quad (9a)$$

$$\text{subject to} \quad \Psi \text{diag}(\bar{\omega}) \Psi^\top - \gamma \mathbf{I}_{N-D-1} \succeq 0 \quad (9b)$$

$$\|\bar{B} \text{diag}(\bar{\omega}) \bar{B}^\top\|_2 \leq \beta \quad (9c)$$

$$E\bar{\omega} = \mathbf{0} \quad (9d)$$

where  $\alpha > 0$  is the weighting parameter of the objectives and  $\beta > 0$  upper bounds the spectral norm of  $\Omega$ . We also set a lower bound  $\gamma > 0$  for (9b) for two explicit reasons, (a) a strict positive definite constraint  $\Psi \text{diag}(\bar{\omega}) \Psi^\top \succ 0$  is numerically infeasible, and (b) when the L1 term is dominating, we want to avoid obtaining the trivial solution  $\bar{\omega} = \mathbf{0}$ .

As a summary of the proposed formulation, with the help of (6e), constraints (6c) and (6d) in  $\mathcal{P}_0$  are convexified using (9b), and (6e) itself is rewritten with (9d) using only the variable  $\bar{\omega}$ . Constraint (9c) combined with the objective (8) achieves the eigenvalue maximization. We now investigate the relationships among these hyperparameters in the following section.

#### D. Choice of Hyperparameters

The need for  $\beta > \gamma$  is straightforward to ensure feasibility since they represent the largest and smallest eigenvalues, respectively. Additionally,  $\beta$  and  $\gamma$  should be sufficiently apart for a relatively large feasible region. The choice of  $\alpha$  determines the objective function, which contains a sparsity term  $\|\bar{\omega}\|_1$  (geometrically a cone) and a convergence term  $-\alpha \psi^\top \bar{\omega}$  (a hyperplane). If the convergence term is dominant, i.e.,  $\alpha$  is large, the hyperplane will unfold the cone such that its global minimum vanishes, in which case the solution is minimized at the boundary of the feasible region which is not necessarily a sparse solution. Mathematically, to keep the global minimum of the objective function,  $\mathbf{0}_{\bar{M}}$  should be in the subdifferential of the objective function, i.e.,

$$\mathbf{0}_{\bar{M}} \in \partial(\|\bar{\omega}\|_1 - \alpha \psi^\top \bar{\omega}), \quad (10)$$

which directly yields that all elements of  $\alpha \psi$  are in the range  $(-1, 1)$ . Recalling that all elements of  $\psi$  are nonnegative from the definition  $\psi = \text{diag}(\Psi^\top \Psi)$  and  $\alpha > 0$ , we conclude that the choice of  $\alpha$  that entails a sparse solution is

$$0 < \alpha < \frac{1}{\|\psi\|_\infty}, \quad (11)$$

where the infinity norm is the largest value of  $\psi$ . Since a larger  $\alpha$  promotes faster convergence, we recommend using the upper bound value of (11) before fine-tuning. If  $\alpha$  is chosen substantially beyond the range (11), then the sparsity is traded off for the convergence rate, which we will illustrate in the next section.

#### IV. SIMULATIONS

In this section, we validate our proposed framework through several examples, comparing its performance with state-of-the-art methods such as Lin et al. [14], Xiao et al. [16], and Yang et al. [15]. We demonstrate the effect of varying the hyperparameter  $\alpha$ , with a fixed  $\beta = 1$  and  $\gamma = 0.1$ . Since the approach in [14] is not generative and requires a predefined graph, we first apply some sparse graph construction technique [13], [10] to generate an appropriate structure before designing the stress, which is a widely adopted pipeline for AFC works. It is obvious from (1) and (4) that stress has a scaling ambiguity, hence a scalar gain could amplify the smallest nonzero eigenvalue for an accelerated convergence. However, for a fair comparison of the convergence rate, we normalize all the acquired stress matrices for a maximum eigenvalue of 1. The code for the simulations is available online<sup>1</sup>, in which we used CVXPY [20] to solve the proposed convex optimization.

Our main test case is a nongeneric regular polygon with 10 nodes in  $\mathbb{R}^2$  shown in Fig. 2(a), a common geometric pattern in, e.g., antenna arrays for maximizing aperture. The optimized graphs are shown in Fig. 2 where all methods yield a sparsified solution compared with a complete graph, among which our proposed solution offers the least required edges under the right  $\alpha$  setting (11). Also, the resulting graph becomes denser with a larger  $\alpha$ , which substantiates our previous discussion that the optimal solution will fall on the border of the feasible region. Besides, it is also consistent with the intuition that the sparsity term in (9a) is less dominant with a larger  $\alpha$ , so a less sparse solution is achieved. Another observation from the results is that the proposed framework gives symmetric solutions as compared to the state-of-the-art works, which is a key benefit for robustness and for balancing the network load. This is because the proposed framework does not assume any sparsity pattern like [15] or [13], although symmetry is not explicitly enforced.

<sup>1</sup><https://github.com/asil-lab/zli-sparse-urf>

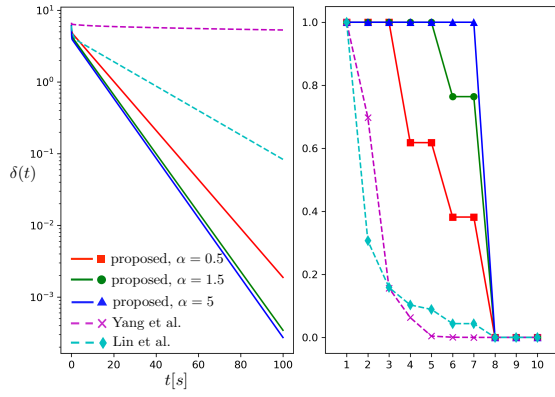


Fig. 3. The convergence of the formation control system (left). The tracking error is  $\delta(t) = \|\mathbf{z}(t) - \mathbf{p}\|_2$ . The normalized eigenvalues of the stress matrices (right). The 7th eigenvalue corresponds to the error convergence speed. Lin et al. is designed over graph Fig. 2 (c).

The convergence of our proposed solution is shown in Fig. 3. We compare the eigenvalues of the acquired stress matrix with the resulting convergence of formation control. When  $\alpha$  is tuned to a larger value, the smallest nonzero eigenvalue and the convergence speed are promoted. The extreme case is  $\alpha = 5$ , where all nonzero eigenvalues are maximized to 1 at the cost of a full connection among nodes in Fig. 2(f). This confirms that the denser the connections, the more efficiently information is spread in the network hence a faster convergence. It is worth noting that our most sparse solution, i.e.,  $\alpha = 0.5$ , still outperforms the existing works.

The generic example from [16] is shown in Fig. 4 where there are 6 nodes in  $\mathbb{R}^2$ . All compared methods present a reduced number of edges from a maximum of  $\bar{M} = 15$  and the proposed technique can reach the lower bound  $\underline{M} = 2N - 2 = 10$  for generic configurations [13]. Additionally, the proposed solution entails the highest  $\lambda_4(\Omega)$ , which is the smallest nonzero eigenvalue.

## V. CONCLUSION

In this work, we formulated the stress matrix design for a universally rigid framework as a convex optimization problem, aiming to minimize the number of edges required for network connectivity while enhancing convergence speed. We provided insights into the selection of hyperparameters, and numerical results demonstrated that our method outperforms existing approaches in both sparsity and convergence speed across our test cases. In future work, we plan to benchmark our solution across broader scenarios with more evaluation criteria such as control energy. We also aim to incorporate additional constraints, such as distance constraints and balanced load distribution, to refine our network design further.

## REFERENCES

- [1] K.-K. Oh, M.-C. Park, and H.-S. Ahn, "A survey of multi-agent formation control," *Automatica*, vol. 53, pp. 424–440, 2015.
- [2] Z. Li and R. T. Rajan, "Geometry-aware distributed kalman filtering for affine formation control under observation losses," in *2023 26th International Conference on Information Fusion (FUSION)*. IEEE, 2023, pp. 1–7.

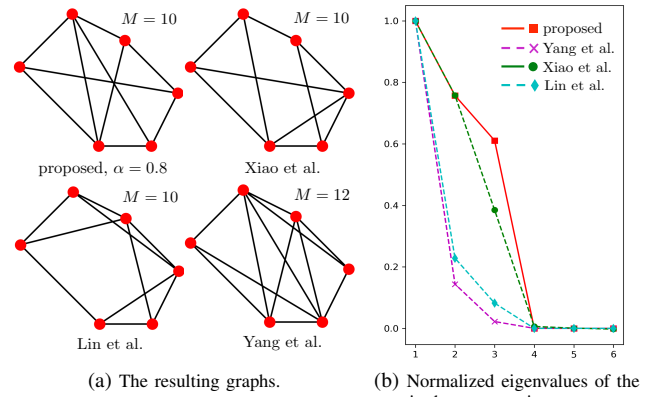


Fig. 4. Comparison of results using the generic configuration from [16]. A structure from [10] is used for Lin et al.

- [3] M. Van Der Marel and R. T. Rajan, "Distributed kalman filters for relative formation control of multi-agent systems," in *2022 30th European Signal Processing Conference (EUSIPCO)*. IEEE, 2022, pp. 1422–1426.
- [4] A. Y. Alfakih and Y. Ye, "On affine motions and bar frameworks in general position," *Linear Algebra and its Applications*, vol. 438, no. 1, pp. 31–36, 2013.
- [5] R. Heusdens and G. Zhang, "Distributed optimisation with linear equality and inequality constraints using pdmm," *IEEE Transactions on Signal and Information Processing over Networks*, 2024.
- [6] R. T. Rajan, G. Leus, and A.-J. Van Der Veen, "Relative kinematics of an anchorless network," *Signal Processing*, vol. 157, pp. 266–279, 2019.
- [7] J. Zheng and A. Jamalipour, *Wireless sensor networks: a networking perspective*. John Wiley & Sons, 2009.
- [8] S. Zhao, "Affine formation maneuver control of multiagent systems," *IEEE Transactions on Automatic Control*, vol. 63, no. 12, pp. 4140–4155, 2018.
- [9] Z. Zhu, A. M.-C. So, and Y. Ye, "Universal rigidity: Towards accurate and efficient localization of wireless networks," in *2010 Proceedings IEEE INFOCOM*. IEEE, 2010, pp. 1–9.
- [10] R. Connelly and S. D. Guest, *Frameworks, tensegrities, and symmetry*. Cambridge University Press, 2022.
- [11] Y. Kim and M. Mesbahi, "On maximizing the second smallest eigenvalue of a state-dependent graph laplacian," *IEEE Transactions on Automatic Control*, vol. 51, no. 1, pp. 116–120, 2006.
- [12] R. Olfati-Saber, J. A. Fax, and R. M. Murray, "Consensus and cooperation in networked multi-agent systems," *Proceedings of the IEEE*, vol. 95, no. 1, pp. 215–233, 2007.
- [13] S. D. Kelly and A. Micheletti, "A class of minimal generically universally rigid frameworks," *arXiv preprint arXiv:1412.3436*, 2014.
- [14] Z. Lin, L. Wang, Z. Chen, M. Fu, and Z. Han, "Necessary and sufficient graphical conditions for affine formation control," *IEEE Transactions on Automatic Control*, vol. 61, no. 10, pp. 2877–2891, 2015.
- [15] Q. Yang, M. Cao, H. Fang, and J. Chen, "Constructing universally rigid tensegrity frameworks with application in multiagent formation control," *IEEE Transactions on Automatic Control*, vol. 64, no. 1, pp. 381–388, 2018.
- [16] F. Xiao, Q. Yang, X. Zhao, and H. Fang, "A framework for optimized topology design and leader selection in affine formation control," *IEEE Robotics and Automation Letters*, vol. 7, no. 4, pp. 8627–8634, 2022.
- [17] K. B. Petersen, M. S. Pedersen et al., "The matrix cookbook," *Technical University of Denmark*, vol. 7, no. 15, p. 510, 2008.
- [18] R. Connelly, "Generic global rigidity," *Discrete & Computational Geometry*, vol. 33, pp. 549–563, 2005.
- [19] S. J. Gortler and D. P. Thurston, "Characterizing the universal rigidity of generic frameworks," *Discrete & Computational Geometry*, vol. 51, no. 4, pp. 1017–1036, 2014.
- [20] S. Diamond and S. Boyd, "CVXPY: A Python-embedded modeling language for convex optimization," *Journal of Machine Learning Research*, vol. 17, no. 83, pp. 1–5, 2016.