Citation: M. Aranda, G. López-Nicolás and Y. Mezouar. Distributed Linear Control of Multirobot Formations Organized in Triads. IEEE Robotics and Automation Letters, vol. 6, no. 4, pp. 310-317, Oct. 2021

Distributed Linear Control of Multirobot Formations Organized in Triads

Miguel Aranda[1], Gonzalo López-Nicolás[2], and Youcef Mezouar[1]

***Abstract*—This letter addresses the problem of controlling multiple robots to form a prescribed team shape in two-dimensional space. We consider a team organization in interlaced triads (i.e., groups of three robots). For each triad we define a measure of geometric deformation relative to its prescribed shape. Our main contribution is a novel distributed control law, defined as the gradient descent on the sum of these triangular deformation measures. We show that this geometrically motivated control law is linear, and bears analogies with existing formulations. Moreover, in comparison with these formulations our controller is simpler and more flexible to design, converges to the globally optimal shape by construction, and allows analysis of the team size dynamics. We illustrate the proposed approach in simulation.**

***Index Terms*—Multi-robot systems, distributed robot systems, autonomous agents, formation control.**

I. Introduction

FORMATION control [1] encompasses various problems that involve a team of robots remaining close to a prescribed geometric reference. Well-established methodologies exist based, e.g., on distances [2]–[4], relative positions [5]–[7], and bearings [8]–[10]. In other cases, the desired geometric arrangement is encoded by, e.g., moments [11], or it emerges implicitly from the robots' motion policies [12]. One particular formation control problem is that of achieving a prescribed pattern of robot positions up to rotation, translation, and uniform scaling. We refer to this problem as *shape control*. A popular approach for shape control in two dimensions based on a complex-Laplacian formulation was introduced in [13], and has later been applied in various scenarios (with, e.g., different robot dynamics and interconnection topologies) in [14]–[18]. Henceforth, we use the term *Laplacian shape control* to refer collectively to the works employing this approach. Laplacian shape control uses a linear formulation and it rests mainly on graph-theoretic and algebraic insights. The control law for each robot is not based on regulating the values of the local geometric variables (relative positions of graph neighbors) that the robot measures. Rather, the design requires the centralized computation of a global gain/weight system matrix (henceforth called *control matrix*).

The motivation for this letter is that minimizing deformation (a concept based on *optimal* shape alignment) is an efficient shape control strategy. We consider a robot team with a formation graph composed by interlaced *triads* (i.e., groups of three robots). Due to their interesting properties, triads have already been used in multirobot control works [4], [19]–[21]. The graph topology we propose is distributed, naturally well-adapted to encode proximity-based interactions, and modular. For each triad, we define a measure of its deformation relative to its desired shape. We propose to perform gradient descent on a cost function equal to the sum of these triangular deformation measures. The resulting control law only requires each robot to measure the relative positions of its graph neighbors, in its local independent coordinate frame. We show that this geometrically motivated approach is effectively a linear shape control. In this respect, our approach is analogous to Laplacian shape control and it enjoys the advantages of linear formulations. But due to its geometric grounding, our method provides these three contributions, not found in the existing Laplacian shape control works: (i) It has optimality (i.e., convergence to the optimum in least-squares sense) by construction. (ii) It allows analysis of the team size dynamics; in particular, we show the size is non-increasing, and its reduction is proportional to the deformation. (iii) It does not require the computation of a control matrix, thus providing a simpler and more flexible controller design.

Many approaches for controlling a formation of an arbitrary number of robots assume the desired formation has a prescribed physical size [2]–[6]. These works consider different conditions in terms of, e.g., measured variables, controlled variables, and reference frames. For the shape control problem (i.e., when the size of the formation is not prescribed) Laplacian shape control [13]–[18] is a powerful approach due to its interesting properties: linearity, global stability, no need for the robots to have a common sense of orientation, no need for communications. The method we propose maintains these properties. Bearing-based methods (surveyed in [22]) can solve the shape control problem, with the important feature of using only bearing measurements. Compared to Laplacian shape control, however, the existing globally stable bearing-based methods require the robots either to have a common sense of orientation (e.g., [9], [10]), or to use communications to share measurements (e.g., [8]). They also lack the advantages associated with linear control formulations. Finally, other related works have exploited least-

Manuscript received: June, 17, 2021; Accepted August, 4, 2021.

This paper was recommended for publication by Editor M. Ani Hsieh upon evaluation of the Associate Editor and Reviewers' comments. This work was supported by projects COMMANDIA (SOE2/P1/F0638), which is cofinanced by Interreg Sudoe Programme (European Regional Development Fund); SOFTMANBOT, which received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 869855; and PGC2018-098719-B-I00 (MCIU/AEI/FEDER, UE).

[1]M. Aranda and Y. Mezouar are with CNRS, Clermont Auvergne INP, Institut Pascal, Université Clermont Auvergne, Clermont-Ferrand, France. Email: first_name.last_name@sigma-clermont.fr.

[2]G. López-Nicolás is with Instituto de Investigación en Ingeniería de Aragón, Universidad de Zaragoza, Spain. Email: gonlopez@unizar.es.

Digital Object Identifier (DOI): see top of this page.

squares alignment in formation control, as we do here: [7], [23] address a different problem (control of a formation with prescribed size), propose nonlinear control laws, and require denser formation graphs than us, while [24] proposes a non-distributed, global deformation-based control; here, the control is distributed, i.e., more fault-tolerant and scalable.

Aside from providing practically relevant contributions (optimality, knowledge of size dynamics, design simplicity), our work is also significant theoretically, as we identify a novel link between geometric and linear algebraic descriptions. This link can be useful for future developments. We present formal analysis of the controller, including the proof of its convergence. Due to being gradient-based, our method can withstand perturbations and it admits different realistic kinematic models (e.g., unicycles). It is also well adapted to handle reconfigurations of the formation during execution. We illustrate all these features with simulations.

II. Problem formulation

Definitions and preliminaries: We denote the transpose as $'$. The norm we use is the Euclidean one. $\otimes$ denotes the Kronecker product. $\mathbf{I}_N$ is the $N \times N$ identity matrix. $\mathbf{1}_N$ is a column vector of N ones. For mean-subtraction operations we will use a *centering matrix* $\mathbf{K} \in \mathbb{R}^{2N \times 2N}$, defined as $\mathbf{K} = (\mathbf{I}_N - (1/N)\mathbf{1}_N\mathbf{1}_N') \otimes \mathbf{I}_2$. We define $\mathbf{S} = [(0,1)', (-1,0)']$, i.e., a counterclockwise rotation of $\pi/2$ rad, and $\mathbf{T} = \mathbf{I}_N \otimes \mathbf{S}$. When it is clear from the context, we do not always notate time dependence of variables (t). $\cdot(t_a)$ means $\cdot$ at time $t = t_a$. We consider a set of robots $\mathcal{N} = \{1, 2, ..., N\}$, $N > 2$. For each robot i, we define its position as $\mathbf{q}_i \in \mathbb{R}^2$, expressed in an arbitrary reference frame. We specify a prescribed *desired shape* as a geometric layout with a position $\mathbf{c}_i$ for each robot. It is assumed that no two $\mathbf{c}_i$ are identical. We encode the current positions and the desired shape specification as column vectors $\mathbf{q} = [\mathbf{q}_1', \mathbf{q}_2', ..., \mathbf{q}_N']'$, $\mathbf{c} = [\mathbf{c}_1', \mathbf{c}_2', ..., \mathbf{c}_N']'$. The *centroid* of a point set $\{\mathbf{v}_i, \forall i \in \mathcal{N}\}$ is defined as $\mathbf{v}_0 = (1/N)\sum_{i\in\mathcal{N}} \mathbf{v}_i \in \mathbb{R}^2$. Without any loss of generality and for simplicity, we assume $\mathbf{c}_0 = \mathbf{0} \in \mathbb{R}^2$. We define the *size* of the team, $w \in \mathbb{R}^{\geq 0}$, as the quadratic mean of the robots' distances to the centroid:

$$w = \sqrt{(1/N)\sum_{i\in\mathcal{N}} ||\mathbf{q}_i - \mathbf{q}_0||^2} = \sqrt{1/N}||\mathbf{K}\mathbf{q}||. \quad (1)$$

Graph topology: We model robot interactions via a static undirected graph $\mathcal{G} = \{\mathcal{N}, \mathcal{E}\}$ with vertex set $\mathcal{N}$ and edge set $\mathcal{E}$. Each vertex corresponds to a robot (we will refer to vertex/robot indistinctly), and each edge to a control interaction between two robots. We assume that $\mathcal{G}$ is structured in triads. A triad $\mathcal{T} = \{i_1, i_2, i_3\}$ is a set of three vertices with edges between every pair. Let us assume there are M distinct triads in $\mathcal{G}$. We call the set of triads $\mathcal{M} = \{\mathcal{T}_1, \mathcal{T}_2, ..., \mathcal{T}_M\}$. Let us define an undirected *triad graph*, $\mathcal{G}_t = \{\mathcal{M}, \mathcal{E}_t\}$ where each vertex is a triad and there is an edge between two vertices $\mathcal{T}_m$ and $\mathcal{T}_n$ if and only if $|\mathcal{T}_m \cap \mathcal{T}_n| = 2$, i.e., the two triads share exactly two robots. Then, we define $\mathcal{G}$ as any graph that satisfies simultaneously the following two conditions:

- GC1: $\forall\ i \in \mathcal{N}$, $\exists\ \mathcal{T}_m \in \mathcal{M} \mid i \in \mathcal{T}_m$.
- GC2: $\mathcal{G}_t$ is connected.

In plain words, GC1 means every robot belongs to at least one triad, while GC2 means the triads are interlaced. These conditions will allow us to ensure team shape convergence. Note that this specification includes, as a particular case, the graphs called *triangulated Laman graphs* in [4]. These are a class of minimally rigid graphs in $\mathbb{R}^2$.

Our motivations for choosing this graph topology are: (i) We are interested in using the concept of deformation, and a triad is the minimal grouping of robots for which such concept can be defined meaningfully. (ii) Although Laplacian shape control can work with certain classes of non-rigid graphs (2-rooted), requiring as we do a minimally rigid graph is still commensurate with very many formation control works (e.g., the distance-based family [2]–[4]). (iii) Triangulations (e.g., triangle meshes or lattices) are a well-known efficient way to define proximity-based interactions in geometrical structures; triangular formations of the type we consider have been used in multirobot control [4], [19]–[21]. (iv) Triads have high modularity, which facilitates scalability: a new robot can be added to a formation with simple rules and in a fully distributed manner.

Robot requirements: Each robot i is modelled as a kinematic point mass that can displace according to the first-order model $\dot{\mathbf{q}}_i = \mathbf{u}_i$, where $\mathbf{u}_i$ is the control input. $\mathbf{u} = [\mathbf{u}_1', \mathbf{u}_2', ..., \mathbf{u}_N']' \in \mathbb{R}^{2N}$ expresses the input for the full team. Extensions to robots with further motion constraints (saturated input, unicycle kinematics) are also illustrated, in the simulations (Sect. V). Robot i needs to know what triads it belongs to, and what their desired shape is. Robot i needs to measure (e.g., with an onboard vision sensor) the relative positions of all its neighbors in $\mathcal{G}$. i can express these position vectors in an arbitrarily centered and oriented coordinate frame, completely independent from the other robots' frames. Wireless communication among neighboring robots would be needed for high-level coordination purposes (e.g., formation reconfiguration); however, importantly, i can compute the control law we propose using solely its own measurements, with no need for communications with the other robots.

Control goal: The goal we consider is for the robots' positions $\mathbf{q}$ to form the shape encoded in $\mathbf{c}$. Using a common definition of the concept of shape of a set of points [25], this implies that these two sets of positions must be equal up to translation, rotation and uniform scaling. We call this the *Shape Condition* (SC). We define the SC by any of the following three expressions, which can be shown to be equivalent to one another:

$$\mathbf{q}_i - \mathbf{q}_j = \mathbf{H}(\mathbf{c}_i - \mathbf{c}_j), \quad \forall i \in \mathcal{N},\ \forall j \in \mathcal{N} \quad (2)$$

$$\mathbf{q}_i - \mathbf{q}_0 = \mathbf{H}\mathbf{c}_i, \quad \forall i \in \mathcal{N} \quad (3)$$

$$\mathbf{q} - \mathbf{1}_N \otimes \mathbf{q_0} = (\mathbf{I}_N \otimes \mathbf{H})\mathbf{c}, \quad (4)$$

where $\mathbf{H}$ is a transformation matrix with the following form:

$$\mathbf{H} = \begin{bmatrix} h_a & -h_b \\ h_b & h_a \end{bmatrix}, \quad h_a \in \mathbb{R},\ h_b \in \mathbb{R}. \quad (5)$$

Any $\mathbf{H} \neq \mathbf{0}$ having this form encodes a shape-preserving transformation that performs rotation and (positive) uniform

scaling. Note that our general definition of the SC includes the case $\mathbf{H} = \mathbf{0}$. Later on in the letter we discuss specifically this particular case, which will not occur in practice. Also, we recall that we consider $\mathbf{c}_0 = \mathbf{0}$. Note that $\mathbf{H}$ does not need to encode translation, as the SC above is formulated subtracting the mean (centroid) from both $\mathbf{q}$ and $\mathbf{c}$.

III. Multirobot shape controller

Two cost functions are used to define and analyze the strategy we propose to achieve the stated control goal.

Global cost function: We call it γ_g and it is based on a least-squares shape alignment: concretely, assume the positions in the specification of the desired shape are translated, rotated, and uniformly scaled so that they align with minimum overall squared error with the current positions; our γ_g is equal to this error. Notice that this error expresses purely the difference *in shape* between the two sets. A usual meaning of the term deformation is the change of shape relative to a reference. This is why we call γ_g a *measure of the deformation* of the current shape relative to the desired one. The reader is referred to [24] for more details on this global cost function (which is called γ in that paper) and to [26] for further discussion of the least-squares alignment problem we consider. We thus define:

$$\gamma_g = \frac{1}{2}\sum_{i\in\mathcal{N}} ||\mathbf{q}_i - \mathbf{q}_0 - \mathbf{H}_g\mathbf{c}_i||^2. \tag{6}$$

$\mathbf{H}_g \in \mathbb{R}^{2\times 2}$ in (6) is a transformation matrix constrained to have the form of $\mathbf{H}$ (5) and which aligns with least-squares error the sets of positions $\{\mathbf{q}_i\}$ and $\{\mathbf{c}_i\}$, $\forall i \in \mathcal{N}$, after subtracting their centroids. $\mathbf{H}_g$ can be obtained via differentiation to find the optimum. Specifically, it has the form (calling $c_s = \sum_{i\in\mathcal{N}} \mathbf{c}_i'\mathbf{c}_i$):

$$\mathbf{H}_g = \begin{bmatrix} h_{a_g} & -h_{b_g} \\ h_{b_g} & h_{a_g} \end{bmatrix},\ h_{a_g} = (1/c_s)\textstyle\sum_{i\in\mathcal{N}}(\mathbf{q}_i - \mathbf{q}_0)'\mathbf{c}_i \\ h_{b_g} = (1/c_s)\textstyle\sum_{i\in\mathcal{N}}(\mathbf{q}_i - \mathbf{q}_0)'\mathbf{S}\mathbf{c}_i. \tag{7}$$

Note that γ_g depends only on the robots' current positions, $\mathbf{q}$, and the desired shape specification, $\mathbf{c}$. $\mathbf{H}_g$ at time t applied on $\mathbf{c}$ defines an optimally rotated and scaled team configuration, which we call $\mathbf{p}(t)$. This configuration will be important in our later analysis. It has the following expression:

$$\mathbf{p}(t) = (\mathbf{I}_N \otimes \mathbf{H}_g(t))\mathbf{c}. \tag{8}$$

Lemma 1: The SC holds if and only if $\gamma_g = 0$.

Proof: (i) Assume $\gamma_g = 0$. This means $\mathbf{q}_i - \mathbf{q}_0 = \mathbf{H}_g\mathbf{c}_i$ $\forall i \in \mathcal{N}$, so the SC holds (3) for $\mathbf{H} = \mathbf{H}_g$. (ii) Suppose the SC holds for a certain $\mathbf{H} = [(h_a, h_b)', (-h_b, h_a)']$. This implies, breaking down in (3) the product $\mathbf{H}\mathbf{c}_i$ into two addends, that $\mathbf{q}_i - \mathbf{q}_0 = h_a\mathbf{c}_i + h_b\mathbf{S}\mathbf{c}_i$, $\forall i \in \mathcal{N}$. We can then see (notice that $(\mathbf{S}\mathbf{c}_i)'\mathbf{c}_i = 0$) that $h_{a_g} = (1/c_s)\sum_{i\in\mathcal{N}}(h_a\mathbf{c}_i + h_b\mathbf{S}\mathbf{c}_i)'\mathbf{c}_i = h_a$. Similarly, $h_{b_g} = h_b$. Hence, from (6), $\gamma_g = 0$. ■

Distributed cost function: We call it γ_d and it is the sum of a deformation measure for every triad. Each such measure (which we call γ_m) is exactly the same as γ_g, but defined for the three robots in the triad (instead of for all the N robots):

$$\gamma_d = \sum_{\mathcal{T}_m\in\mathcal{M}} \gamma_m,\ \gamma_m = \frac{1}{2}\sum_{i\in\mathcal{T}_m} ||(\mathbf{q}_i - \mathbf{q}_{0_m}) - \mathbf{H}_m(\mathbf{c}_i - \mathbf{c}_{0_m})||^2, \tag{9}$$

where:

$$\mathbf{H}_m = \begin{bmatrix} h_{a_m} & -h_{b_m} \\ h_{b_m} & h_{a_m} \end{bmatrix},\ h_{a_m} = \tfrac{1}{c_{sm}}\textstyle\sum_{i\in\mathcal{T}_m}(\mathbf{q}_i - \mathbf{q}_{0_m})'(\mathbf{c}_i - \mathbf{c}_{0_m}) \\ h_{b_m} = \tfrac{1}{c_{sm}}\textstyle\sum_{i\in\mathcal{T}_m}(\mathbf{q}_i - \mathbf{q}_{0_m})'\mathbf{S}(\mathbf{c}_i - \mathbf{c}_{0_m})$$

$$\mathbf{q}_{0_m} = (1/3)\sum_{i\in\mathcal{T}_m}\mathbf{q}_i,\ \ \mathbf{c}_{0_m} = (1/3)\sum_{i\in\mathcal{T}_m}\mathbf{c}_i \\ c_{sm} = \sum_{i\in\mathcal{T}_m}(\mathbf{c}_i - \mathbf{c}_{0_m})'(\mathbf{c}_i - \mathbf{c}_{0_m}). \tag{10}$$

Lemma 2: The SC holds if and only if $\gamma_d = 0$.

Proof: (i) Assume $\gamma_d = 0$. Thus $\gamma_m = 0$, i.e., $\mathbf{q}_i - \mathbf{q}_{0_m} = \mathbf{H}_m(\mathbf{c}_i - \mathbf{c}_{0_m})$, $\forall i \in \mathcal{T}_m$, $\forall \mathcal{T}_m \in \mathcal{M}$, and therefore $\mathbf{q}_i - \mathbf{q}_j = \mathbf{H}_m(\mathbf{c}_i - \mathbf{c}_j)$, $\forall i, j \in \mathcal{T}_m$, $\forall \mathcal{T}_m \in \mathcal{M}$. Consider two given triads $\mathcal{T}_m$ and $\mathcal{T}_n$, $\in \mathcal{M}$, such that they are neighbors in $\mathcal{G}_t$. This means there are two distinct robots (call them i, j) that belong to both $\mathcal{T}_m$ and $\mathcal{T}_n$. We then have $\mathbf{q}_i - \mathbf{q}_j = \mathbf{H}_m(\mathbf{c}_i - \mathbf{c}_j)$ and $\mathbf{q}_i - \mathbf{q}_j = \mathbf{H}_n(\mathbf{c}_i - \mathbf{c}_j)$. Due to the form of the transformation matrices $\mathbf{H}_m$ and $\mathbf{H}_n$, and recalling that $\mathbf{c}_i - \mathbf{c}_j \neq \mathbf{0}$, this implies $\mathbf{H}_m = \mathbf{H}_n$. Hence any two neighbor vertices (triads) in $\mathcal{G}_t$ have the same transformation. Since $\mathcal{G}_t$ is connected (GC2), there is a path (i.e., a sequence of vertices, each being a neighbor of the previous one) between every two vertices in $\mathcal{G}_t$; this means we can use the reasoning above in a transitive manner and cover all vertices of $\mathcal{G}_t$ (i.e., all the M triads), and hence conclude $\mathbf{H}_1 = \mathbf{H}_2 = ... = \mathbf{H}_M = \mathbf{H}_c$ for a single transformation matrix $\mathbf{H}_c$. As every robot is part of a triad (GC1), it follows that for every pair $k \in \mathcal{N}$, $l \in \mathcal{N}$, $\mathbf{q}_k - \mathbf{q}_l = \mathbf{H}_c(\mathbf{c}_k - \mathbf{c}_l)$. Thus, the SC holds (2) for $\mathbf{H} = \mathbf{H}_c$. (ii) Suppose now the SC holds. It is direct to see that this implies, for every triad $\mathcal{T}_m$, $\mathbf{q}_i - \mathbf{q}_{0_m} = \mathbf{H}(\mathbf{c}_i - \mathbf{c}_{0_m})$ $\forall i \in \mathcal{T}_m$. Applying, on every $\mathcal{T}_m$, the same reasoning as in part (ii) of the proof of Lemma 1 leads to $\mathbf{H}_m = \mathbf{H}$, and therefore $\gamma_m = 0$. Hence, $\gamma_d = 0$. ■

A. Control law

Our idea is to achieve the SC via gradient descent on γ_d. Note that Lemma 2 supports the appropriateness of this idea. Hence, each robot i moves following the negative gradient of γ_d with respect to $\mathbf{q}_i$. As $\mathbf{H}_m$ is a minimizer of γ_m and thus the gradient of γ_m with respect to $\mathbf{H}_m$ is zero, the control law for every robot $i \in \mathcal{N}$, $\mathbf{u}_i \in \mathbb{R}^2$, is simply:

$$\mathbf{u}_i = \sum_{\mathcal{T}_m | \, i\in\mathcal{T}_m} \mathbf{u}_{i_m},\quad \mathbf{u}_{i_m} = \mathbf{H}_m(\mathbf{c}_i - \mathbf{c}_{0_m}) - (\mathbf{q}_i - \mathbf{q}_{0_m}). \tag{11}$$

It is useful to interpret $\mathbf{u}_{i_m}$ as the sum of two vectors: one going to the centroid ($\mathbf{q}_{0_m} - \mathbf{q}_i$) and another going from the centroid to the position of i in the optimal rotated and scaled shape for triad $\mathcal{T}_m$, ($\mathbf{H}_m(\mathbf{c}_i - \mathbf{c}_{0_m})$). Hence, the three endpoints of $\mathbf{u}_{i_m}$ for $i \in \mathcal{T}_m$ form the optimal desired shape for that triad. Figure 1 illustrates the sum of partial triad vectors that gives rise to the control law (11).

The control law (11) is fully distributed: what a robot i needs to know online is only the relative positions of its neighbors in $\mathcal{G}$. Consider a triad $\mathcal{T}_m = \{i, j, k\}$. If robot i can measure (e.g., with an onboard sensor) vectors $\mathbf{q}_j - \mathbf{q}_i$ and $\mathbf{q}_k - \mathbf{q}_i$, then it directly knows $\mathbf{q}_j - \mathbf{q}_k$ too. It is easy

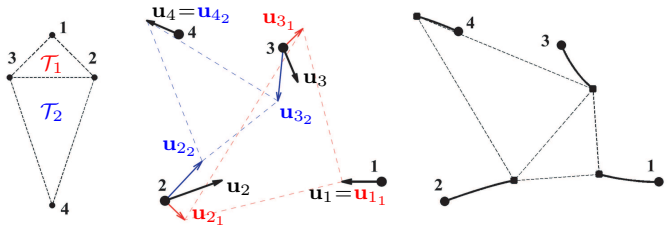

Fig. 1. Controller illustration with $N = 4$ and two triads $\mathcal{T}_1 = \{1, 2, 3\}$, $\mathcal{T}_2 = \{2, 3, 4\}$. Left: Robots' desired geometry and graph edges. Center: Initial robot positions (circles) and control vectors (arrows). Vector endpoints (representing the optimal triangular shapes) shown as triangles in dashed lines. Note the different sizes of the two triangles. Right: robot paths when running the control. Final positions marked as squares.

to see that with this information i can compute the vectors $\mathbf{q}_l - \mathbf{q}_{0_m}$ $\forall l \in \mathcal{T}_m$ and, similarly, $\forall \mathcal{T}_m \mid i \in \mathcal{T}_m$. These are the vectors it needs for the control law, see (11) and (10). To see that i can use an arbitrarily centered and oriented local reference frame, first note that all vectors used are *relative* positions (so they are the same regardless of the origin of the coordinate system). Then, assume there is a rotation by angle θ_i between i's frame and the global frame, with $\mathbf{R}_i(\theta_i) \in SO(2)$ the corresponding matrix. Let us notate for short $\mathbf{q}_{lm} = \mathbf{q}_l - \mathbf{q}_{0_m}$, $\mathbf{c}_{lm} = \mathbf{c}_l - \mathbf{c}_{0_m}$, and use a superscript i for quantities measured in i's frame. Thus $\mathbf{q}^i_{lm} = \mathbf{R}_i \mathbf{q}_{lm}$. Then:

$$h^i_{a_m} = \frac{1}{c_{sm}} \sum_{l=i,j,k} (\mathbf{R}_i \mathbf{q}_{lm})' \mathbf{c}_{lm} = \frac{1}{c_{sm}} \sum_{l=i,j,k} \cos(\theta_i) \mathbf{q}_{lm}{}' \mathbf{c}_{lm} - \sin(\theta_i) \mathbf{q}_{lm}{}' \mathbf{S} \mathbf{c}_{lm} = \cos(\theta_i) h_{a_m} - \sin(\theta_i) h_{b_m},$$

and, analogously: $h^i_{b_m} = \sin(\theta_i) h_{a_m} + \cos(\theta_i) h_{b_m}$. $\mathbf{H}^i_m = \mathbf{R}_i \mathbf{H}_m$ follows directly. Hence, $\mathbf{u}^i_{i_m} = \mathbf{R}_i \mathbf{u}_{i_m}$ in (11). Thus i's motion is the same as when computed in the global frame.

In Laplacian shape control, a control matrix is centrally computed (via numerical optimization) and the resulting control weights are transmitted to the robots. Our purely geometric control law avoids this procedure. As said in Sect. II, what robot i needs to know before running the control is just the identities of its graph neighbors, the triads it belongs to, and the desired shape of those triads. This makes the control design simpler and more scalable. Importantly, it also facilitates reconfiguration of the graph and/or desired shape at a local level. We illustrate this in our simulations.

B. Linear control matrix

The observation that the matrices $\mathbf{H}_m$ in (11) are linear functions of $\mathbf{q}$ prompts us to look for a linear expression of the full dynamical system. We provide this expression next.

Proposition 1: Defining $\mathbf{A}_d, \mathbf{A}_m, \mathbf{L}_m \in \mathbb{R}^{2N \times 2N}$, $\mathbf{L}_{gm} \in \mathbb{R}^{N \times N}$ as follows:

$$\mathbf{A}_d = \sum_{\mathcal{T}_m \in \mathcal{M}} \mathbf{A}_m, \quad \mathbf{A}_m = \frac{\mathbf{L}_m (\mathbf{c}\mathbf{c}' + \mathbf{T}\mathbf{c}\mathbf{c}'\mathbf{T}') \mathbf{L}_m}{\mathbf{c}' \mathbf{L}_m \mathbf{c}} - \mathbf{L}_m, \tag{12}$$

$$\mathbf{L}_m = \mathbf{L}_{gm} \otimes \mathbf{I}_2, \; \mathbf{L}_{gm}[i,j] = \begin{cases} 2/3, & \text{if } i = j \;\&\; i \in \mathcal{T}_m \\ -1/3, & \text{if } i \neq j \;\&\; i,j \in \mathcal{T}_m \\ 0 & \text{otherwise,} \end{cases}$$

the team control law can be expressed as $\mathbf{u} = \dot{\mathbf{q}} = \mathbf{A}_d \mathbf{q}$.

Proof: Let us first express $\mathbf{u}$ as the sum of contributions due to each triad. For this, let us define $\mathbf{u}_{i_m} = \mathbf{0}$ if $i \notin \mathcal{T}_m$, and thus $\mathbf{u}_m = [\mathbf{u}'_{1_m}, \mathbf{u}'_{2_m}, ..., \mathbf{u}'_{N_m}]', \in \mathbb{R}^{2N}$ as the collective dynamics due to triad $\mathcal{T}_m$. Then, $\mathbf{u} = \sum_{\mathcal{T}_m \in \mathcal{M}} \mathbf{u}_m \in \mathbb{R}^{2N}$. Notice that the terms appearing in (11) for the vector $\mathbf{u}_{i_m}$ in a given triad $\mathcal{T}_m = \{i, j, k\}$ can be expressed as:

$$\mathbf{H}_m(\mathbf{c}_i - \mathbf{c}_{0_m}) = (h_{a_m} \mathbf{I}_2 + h_{b_m} \mathbf{S})(\mathbf{c}_i - \mathbf{c}_{0_m}) \tag{13}$$

$$\mathbf{q}_i - \mathbf{q}_{0_m} = (1/3)((\mathbf{q}_i - \mathbf{q}_j) + (\mathbf{q}_i - \mathbf{q}_k)) \tag{14}$$

$$\mathbf{c}_i - \mathbf{c}_{0_m} = (1/3)((\mathbf{c}_i - \mathbf{c}_j) + (\mathbf{c}_i - \mathbf{c}_k)). \tag{15}$$

Observe that these last two expressions are analogous to those of a standard linear consensus algorithm, with its scope restricted to triad $\mathcal{T}_m$. Thus, using the provided matrix $\mathbf{L}_m$, the collective dynamics takes the form:

$$\mathbf{u}_m = h_{a_m} \mathbf{c}_m + h_{b_m} \mathbf{c}^\perp_m - \mathbf{L}_m \mathbf{q}, \tag{16}$$

where we define $\mathbf{c}_m = \mathbf{L}_m \mathbf{c}$, $\mathbf{c}^\perp_m = \mathbf{L}_m \mathbf{T} \mathbf{c}$. Let us now provide vector expressions for the transformation (10). Notice that all terms with form $\mathbf{q}'_{0_m}(\mathbf{c}_i - \mathbf{c}_{0_m})$ and $\mathbf{q}'_{0_m} \mathbf{S}(\mathbf{c}_i - \mathbf{c}_{0_m})$ can be removed because they cancel out when added up $\forall i \in \mathcal{T}_m$. We thus have:

$$h_{a_m} = \frac{\mathbf{q}' \mathbf{c}_m}{\mathbf{c}'_m \mathbf{c}_m}, \quad h_{b_m} = \frac{\mathbf{q}' \mathbf{c}^\perp_m}{\mathbf{c}'_m \mathbf{c}_m}.$$

Using $(\mathbf{q}' \mathbf{c}_m) \mathbf{c}_m = \mathbf{c}_m \mathbf{c}'_m \mathbf{q}$ and $(\mathbf{q}' \mathbf{c}^\perp_m) \mathbf{c}^\perp_m = \mathbf{c}^\perp_m \mathbf{c}^{\perp\prime}_m \mathbf{q}$:

$$h_{a_m} \mathbf{c}_m = \frac{\mathbf{c}_m \mathbf{c}'_m \mathbf{q}}{\mathbf{c}'_m \mathbf{c}_m}, \quad h_{b_m} \mathbf{c}^\perp_m = \frac{\mathbf{c}^\perp_m \mathbf{c}^{\perp\prime}_m \mathbf{q}}{\mathbf{c}'_m \mathbf{c}_m}.$$

We can now combine these expressions with (16) to find:

$$\mathbf{u}_m = \mathbf{A}_m \mathbf{q}, \quad \mathbf{A}_m = \frac{\mathbf{L}_m (\mathbf{c}\mathbf{c}' + \mathbf{T}\mathbf{c}\mathbf{c}'\mathbf{T}') \mathbf{L}_m}{\mathbf{c}' \mathbf{L}_m \mathbf{c}} - \mathbf{L}_m, \tag{17}$$

noting that $\mathbf{L}'_m = \mathbf{L}_m$ and $\mathbf{L}'_m \mathbf{L}_m = \mathbf{L}_m$. Then, recalling $\mathbf{u} = \sum_{\mathcal{T}_m \in \mathcal{M}} \mathbf{u}_m$, the stated result follows. ■

$\mathbf{A}_d$ depends purely on the definition of the desired geometry ($\mathbf{c}$) and of the graph (triad structure). From (12) one can see that $\mathbf{A}_d$ is symmetric. Let us illustrate its structure in more detail using the example in Fig. 1. For that example, $\mathbf{c} = [0, 3/2, 1, 1/2, -1, 1/2, 0, -5/2]'$, $N = 4$, $M = 2$ with $\mathcal{T}_1 = \{1, 2, 3\}$, $\mathcal{T}_2 = \{2, 3, 4\}$. Then, $\mathbf{A}_d = \mathbf{A}_1 + \mathbf{A}_2$, where:

$$\mathbf{A}_1 = \begin{pmatrix} -1/2 & 0 & 1/4 & -1/4 & 1/4 & 1/4 & 0 & 0 \\ 0 & -1/2 & 1/4 & 1/4 & -1/4 & 1/4 & 0 & 0 \\ 1/4 & 1/4 & -1/4 & 0 & 0 & -1/4 & 0 & 0 \\ -1/4 & 1/4 & 0 & -1/4 & 1/4 & 0 & 0 & 0 \\ 1/4 & -1/4 & 0 & 1/4 & -1/4 & 0 & 0 & 0 \\ 1/4 & 1/4 & -1/4 & 0 & 0 & -1/4 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix},$$

$$\mathbf{A}_2 = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -5/12 & 0 & 1/3 & 1/4 & 1/12 & -1/4 \\ 0 & 0 & 0 & -5/12 & -1/4 & 1/3 & 1/4 & 1/12 \\ 0 & 0 & 1/3 & -1/4 & -5/12 & 0 & 1/12 & 1/4 \\ 0 & 0 & 1/4 & 1/3 & 0 & -5/12 & -1/4 & 1/12 \\ 0 & 0 & 1/12 & 1/4 & 1/12 & -1/4 & -1/6 & 0 \\ 0 & 0 & -1/4 & 1/12 & 1/4 & 1/12 & 0 & -1/6 \end{pmatrix}.$$

The zero rows and columns illustrate that the action of each $\mathbf{A}_m$ is restricted to the three robots in $\mathcal{T}_m$. We insist that designing or knowing $\mathbf{A}_d$ is not needed to implement the controller (see Sect. III-A). Let us define $\mathbf{A}_g$, a *global* matrix, analogous to an $\mathbf{A}_m$ matrix but defined on the set $\mathcal{N}$ instead:

$$\mathbf{A}_g = \mathbf{C} - \mathbf{K}, \quad \mathbf{C} = \frac{\mathbf{c}\mathbf{c}' + \mathbf{T}\mathbf{c}\mathbf{c}'\mathbf{T}'}{\mathbf{c}'\mathbf{c}}. \tag{18}$$

Note we can express γ_g and γ_d in terms of these matrices:

$$\gamma_g = -(1/2)\mathbf{q}'\mathbf{A}_g\mathbf{q}, \quad \gamma_d = -(1/2)\mathbf{q}'\mathbf{A}_d\mathbf{q}. \tag{19}$$

IV. Controller analysis

We use in the following analysis the two descriptions (geometric and linear algebraic) provided thus far. We will study the invariance properties (Prop. 2), convergence (Thm. 1) and behavior of the team size (Props. 3 and 4, Corol. 1).

Proposition 2: $\mathbf{q}_0$ and $\mathbf{H}_g$ are invariant under the action of the control law (11).

Proof: To see that $\mathbf{q}_0$ is invariant, note $\dot{\mathbf{q}}_0 = (1/N)(\mathbf{1}_N \otimes \mathbf{I}_2)'\dot{\mathbf{q}}$, which after substitution yields:

$$\dot{\mathbf{q}}_0 = \sum_{\mathcal{T}_m \in \mathcal{M}} (\mathbf{1}_N \otimes \mathbf{I}_2)'\mathbf{L}_m \left(\frac{(\mathbf{cc}' + \mathbf{Tcc}'\mathbf{T}')\mathbf{L}_m}{\mathbf{c}'\mathbf{L}_m\mathbf{c}} - \mathbf{I}_{2N}\right)\frac{\mathbf{q}}{N}, \tag{20}$$

but $(\mathbf{1}_N \otimes \mathbf{I}_2)'\mathbf{L}_m = (\mathbf{1}_N'\mathbf{L}_{gm}) \otimes \mathbf{I}_2 = \mathbf{0}\ \forall \mathcal{T}_m$, since $\mathbf{1}_N'\mathbf{L}_{gm} = \mathbf{0}$. To show that $\mathbf{H}_g$ is also invariant, we will express each component of the matrix in (7) in vector form. For this, notice that all terms with form $\mathbf{q}_0'\mathbf{c}_i$ and $\mathbf{q}_0'\mathbf{S}\mathbf{c}_i$ can be removed because they cancel out when added up $\forall i \in \mathcal{N}$. We then have the following dynamics:

$$\dot{h}_{a_g} = \frac{\dot{\mathbf{q}}'\mathbf{c}}{\mathbf{c}'\mathbf{c}} = \frac{(\mathbf{A}_d\mathbf{q})'\mathbf{c}}{\mathbf{c}'\mathbf{c}} = \frac{\mathbf{q}'\mathbf{A}_d\mathbf{c}}{\mathbf{c}'\mathbf{c}} = 0 \tag{21}$$

$$\dot{h}_{b_g} = \frac{\dot{\mathbf{q}}'\mathbf{T}\mathbf{c}}{\mathbf{c}'\mathbf{c}} = \frac{(\mathbf{A}_d\mathbf{q})'\mathbf{T}\mathbf{c}}{\mathbf{c}'\mathbf{c}} = \frac{\mathbf{q}'\mathbf{A}_d\mathbf{T}\mathbf{c}}{\mathbf{c}'\mathbf{c}} = 0, \tag{22}$$

where we have used that $\mathbf{A}_d$ is symmetric and that $\mathbf{A}_d\mathbf{c} = \mathbf{A}_d\mathbf{T}\mathbf{c} = \mathbf{0}$. To see that $\mathbf{A}_d\mathbf{c} = \mathbf{0}$, note that $\mathbf{A}_d\mathbf{c} = \mathbf{u}(\mathbf{q} = \mathbf{c})$. Here, $\mathbf{q} = \mathbf{c}$ implies the SC, and hence $\gamma_d = 0$ (Lemma 2). Comparing (9) and (11), $\gamma_d = 0$ immediately implies $\mathbf{u}_i = \mathbf{0}$ $\forall i \in \mathcal{N}$, i.e., $\mathbf{u}(\mathbf{q} = \mathbf{c}) = \mathbf{A}_d\mathbf{c} = \mathbf{0}$. Analogously, $\mathbf{A}_d\mathbf{T}\mathbf{c} = \mathbf{0}$ because $\mathbf{T}\mathbf{c}$ is simply $\mathbf{c}$ rotated $\pi/2$ rad., so $\mathbf{q} = \mathbf{T}\mathbf{c}$ implies the SC as well. ■

Lemma 3: Under the action of control (11), $\dot{\gamma}_g = -2\gamma_d$.

Proof: We first provide the following expression:

$$\mathbf{A}_g\mathbf{A}_d = \mathbf{C}\mathbf{A}_d - \mathbf{K}\mathbf{A}_d = -\mathbf{K}\mathbf{A}_d = -\mathbf{A}_d. \tag{23}$$

To obtain it we have used two facts:

(i) $\mathbf{C}\mathbf{A}_d = \mathbf{0}$. To show this fact, note first that $\mathbf{C}\mathbf{A}_d = \mathbf{A}_d\mathbf{C}$, due to symmetry of the matrices involved. We will next show that for every column $j \in \mathcal{N}$ of $\mathbf{C}$, $\mathbf{A}_d col_j(\mathbf{C}) = \mathbf{0}$. Since $\mathbf{A}_d col_j(\mathbf{C}) = \mathbf{u}(\mathbf{q} = col_j(\mathbf{C}))$, we will show $\mathbf{u}(\mathbf{q} = col_j(\mathbf{C})) = \mathbf{0}$. Notice in (18) that every column j has the form $col_j(\mathbf{C}) = a_j\mathbf{c} + b_j\mathbf{T}\mathbf{c} = (\mathbf{I}_N \otimes \mathbf{H}_j)\mathbf{c}$ with $\mathbf{H}_j = a_j\mathbf{I}_2 + b_j\mathbf{S} \in \mathbb{R}^{2\times 2}$. Note that this $\mathbf{H}_j$ has the form (5). Therefore $\mathbf{q} = col_j(\mathbf{C})$ is a configuration $\mathbf{q}$ with zero centroid and which satisfies the SC (4). Hence, from Lemma 2, if $\mathbf{q} = col_j(\mathbf{C})$, $\gamma_d = 0$. Comparing (9) and (11), $\gamma_d = 0$ immediately implies $\mathbf{u}_i = \mathbf{0}$ $\forall i \in \mathcal{N}$, thus $\mathbf{u}(\mathbf{q} = col_j(\mathbf{C})) = \mathbf{0}$ $\forall j \in \mathcal{N}$, i.e., $\mathbf{A}_d\mathbf{C} = \mathbf{C}\mathbf{A}_d = \mathbf{0}$.

(ii) $\mathbf{K}\mathbf{A}_d = \mathbf{A}_d$. From (12), if $\mathbf{K}\mathbf{L}_m = \mathbf{L}_m$ $\forall \mathcal{T}_m \in \mathcal{M}$, then $\mathbf{K}\mathbf{A}_d = \mathbf{A}_d$. To show that $\mathbf{K}\mathbf{L}_m = \mathbf{L}_m$ $\forall \mathcal{T}_m \in \mathcal{M}$, note $\mathbf{K}\mathbf{L}_m = \mathbf{L}_m - (1/N)(\mathbf{1}_N\mathbf{1}_N' \otimes \mathbf{I}_2)\mathbf{L}_m$; here $(\mathbf{1}_N\mathbf{1}_N' \otimes \mathbf{I}_2)\mathbf{L}_m$ is a stacking of $(\mathbf{1}_N' \otimes \mathbf{I}_2)\mathbf{L}_m = (\mathbf{1}_N \otimes \mathbf{I}_2)'\mathbf{L}_m$ which, as noted in the proof of Prop. 2, is zero. This shows fact (ii) and hence (23). Using (23) and the symmetry of $\mathbf{A}_g$:

$$\dot{\gamma}_g = -\mathbf{q}'\mathbf{A}_g\dot{\mathbf{q}} = -\mathbf{q}'\mathbf{A}_g\mathbf{A}_d\mathbf{q} = \mathbf{q}'\mathbf{A}_d\mathbf{q} = -2\gamma_d. \tag{24}$$

■

To obtain the remaining results we use the next expression (from Lemma 2 in [24]), with w from (1) and $\mathbf{p}$ from (8):

$$\gamma_g(t) = \frac{1}{2}(Nw^2(t) - ||\mathbf{p}(t)||^2). \tag{25}$$

Proposition 3: Under the control (11), the size of the team is non-increasing, i.e., $\dot{w}(t) \leq 0\ \forall t$.

Proof: As $\dot{\mathbf{H}}_g(t) = \mathbf{0}$ (Prop. 2), $\dot{\mathbf{p}}(t) = \mathbf{0}$. Using this and $\dot{\gamma}_g \leq 0$ (Lemma 3) in (25), we conclude $\dot{w}(t) \leq 0$. ■

Theorem 1: Using control law (11) the team converges asymptotically from any initial condition to the globally optimal configuration at time zero satisfying the SC.

Proof: Let us choose, without loss of generality and for simplicity of the exposition, the origin of the space $\mathbb{R}^{2N}$ for $\mathbf{q}$ at $\mathbf{q}_0(0)$. Since $\mathbf{q}_0$ is invariant (Prop. 2), we then have that $\mathbf{q}(t)$ is always centered, i.e., $\mathbf{K}\mathbf{q}(t) = \mathbf{q}(t)\ \forall t$, and equivalently $\mathbf{q}_0(t) = \mathbf{0}\ \forall t$. As $||\mathbf{q}(t)|| = ||\mathbf{K}\mathbf{q}(t)|| = \sqrt{N}w(t)$ and $\dot{w}(t) \leq 0$ (Prop. 3), for any $\mathbf{q}(0)$ the system's state $\mathbf{q}$ is guaranteed to remain in $\Omega = \{\mathbf{q} \mid ||\mathbf{q}|| \leq ||\mathbf{q}(0)||\}$. Hence the compact set Ω is positively invariant. Define $V(\mathbf{q}) = \gamma_g$. We have that $\dot{V}(\mathbf{q}) = -2\gamma_d \leq 0$ (Lemma 3). Hence, by virtue of LaSalle's invariance principle, $\mathbf{q}$ converges to the largest invariant set within Ω in which $\dot{V}(\mathbf{q}) = 0$, i.e., in which $\gamma_d = 0$. $\gamma_d = 0$ implies $\gamma_g = 0$, and it implies the SC holds (Lemmas 1 and 2). Since $\gamma_g = 0$ implies (4) with $\mathbf{H} = \mathbf{H}_g$, and recalling $\mathbf{q}_0 = \mathbf{0}$, we have $\lim_{t\to\infty}\mathbf{q}(t) = (\mathbf{I}_N \otimes \mathbf{H}_g(t))\mathbf{c} = \mathbf{p}(t)$. As $\mathbf{H}_g(t) = \mathbf{H}_g(0)\ \forall t$ (Prop. 2), $\lim_{t\to\infty}\mathbf{q}(t) = \mathbf{p}(0)$. This is the optimal, least-squares configuration satisfying the SC at $t = 0$. ■

Remark 1: Suppose we apply a common multiplicative gain $k_u \in \mathbb{R}_{>0}$ on every $\mathbf{u}_i$, which means $\mathbf{u} = \dot{\mathbf{q}} = k_u\mathbf{A}_d\mathbf{q}$. It is direct to see that this only modifies the rate of convergence without affecting the other dynamic properties. ⋄

We already know (Prop. 3) the team has a tendency to shrink. Next, we study more precisely the dynamics of $w(t)$.

Proposition 4: The minimum value of the size over time under the action of the controller (11) satisfies:

$$\min_t w^2(t) = \lim_{t\to\infty} w^2(t) = w^2(0) - \frac{2}{N}\gamma_g(0). \tag{26}$$

Proof: Considering (25) at time $t = 0$ gives $||\mathbf{p}(0)||^2 = Nw^2(0) - 2\gamma_g(0)$. In addition, since the convergence is to the initial optimum (Thm. 1), we have $\lim_{t\to\infty} w(t) = \sqrt{1/N}||\mathbf{p}(0)||$. The stated result follows from these two expressions, and the observation that w is minimum at $t \to \infty$, because $\dot{w}(t) \leq 0$ (Prop. 3). ■

Corollary 1: $w(t) \to 0$ if and only if $\gamma_g(0) = \frac{N}{2}w^2(0)$, which is the maximum possible value of γ_g when $w = w(0)$.

Proof: We ignore the infeasible case $w(0) = 0$. The first statement follows from (26). Then, consider (25); as $||\mathbf{p}(t)||^2 \geq 0$, then for any size w_c, the maximum possible value of γ_g for any configuration with size w_c is $\frac{N}{2}w_c^2$. ■

Remark 2: It is important to note that (i) the size reduction is proportional to the initial deformation $\gamma_g(0)$ (26); a barely deformed team will barely shrink, (ii) $w(t) \to 0$ only happens in *worst-case* configurations in terms of the control objective (i.e., configurations where the deformation measure γ_g is maximum) and (iii) the set of these configurations is measure

zero, so they are irrelevant in practice because an infinitesimal perturbation takes the system out of them. ⋄

The theoretical possibility of the size becoming zero also exists in the Laplacian shape control works. We highlight the fact that here, unlike in these works, we provide a specific analysis of the behavior of the team's size under the linear control law. Methods to control the size to a specific value were proposed in the Laplacian shape control works, via additional *nonlinear* control terms and based on, e.g., a leader-based approach [13], [14] or a concurrent distance-based control law [17]. As our underlying formulation is analogous to [13], [14], [17], we could also use those terms to fix the size.

An illustrative example is shown in Fig. 2. The initial configuration is the desired one reflected with respect to the x axis. This makes the initial deformation maximum: specifically, $\gamma_g(0) = \frac{N}{2}w^2(0) = 4\,\mathrm{m}^2$. Thus, with the control law (11), $\mathbf{H}_g(t) = \mathbf{0}$ and $\mathbf{p}(t) = \mathbf{0}\ \forall t$, and the robots end up meeting at the centroid. We then consider the same initial configuration but make $\mathbf{u}_1(t) = \mathbf{u}_2(t) = \mathbf{0}\ \forall t$ (i.e., we leave robots 1 and 2 static). The team then converges to the desired shape, and the size, orientation and centroid of the final configuration are all fixed by the positions of 1 and 2. Note that this is an example of fixing the size with two leader robots, analogously to what is done in [13].

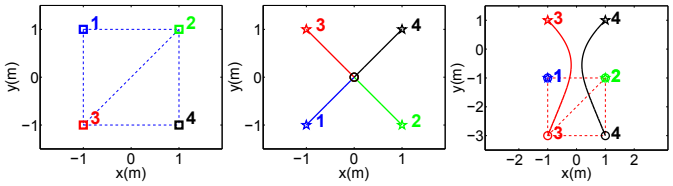

Fig. 2. Example with $N = 4$. Left: Desired geometry. Center: paths with (11). Right: paths with robots 1, 2 static. Initial positions marked as stars, final positions as circles, paths as solid lines, graph edges as dashed lines.

V. Simulation Examples

We present several tests. A **video** is also attached and accessible at `https://youtu.be/3H0C6QOoePk`.

A. Minimally rigid graph and no motion constraints

We consider $N = 8$ and a minimally rigid graph in $\mathbb{R}^2$. The desired shape consists of three chained squares. Note that, interestingly, the control matrix $\mathbf{A}_d$ (12) is exactly the same as the one obtained with the numerical optimization procedure proposed in [16] for Laplacian shape control. That procedure places the eigenvalues of the matrix so as to maximize convergence rate, for a given graph topology. Therefore, our geometric control approach maximizes convergence rate in this example; this interesting fact is reasonable because the approach is based on least-squares. Figure 3 illustrates the results. The centroid remains static with coordinates $x = 10.814\,\mathrm{m}$, $y = -6.201\,\mathrm{m}$. The global matrix $\mathbf{H}_g$ stays constant with values $h_{a_g} = 1.091$, $h_{b_g} = 0.474$. The team converges to the initial global optimum.

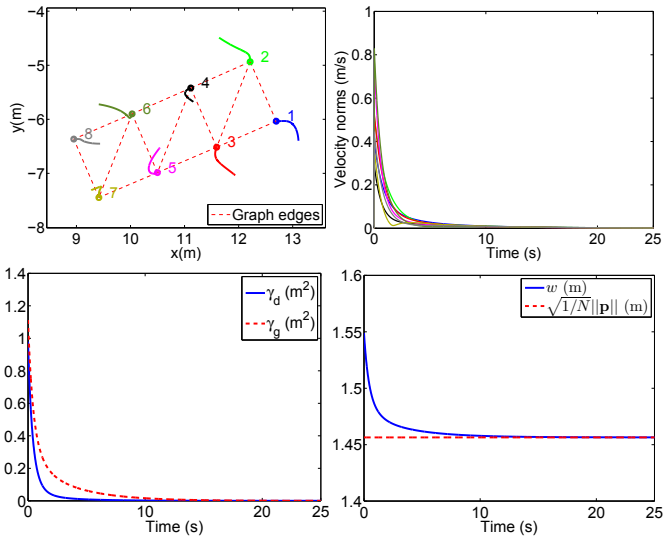

Fig. 3. Results for Sect. V-A. Top: (left) robot paths converging to the final configuration having the desired shape (final positions marked as circles); (right) norms of velocities ($||\mathbf{u}_i||$) over time. Bottom: (left) cost functions over time; (right) current (w) and destination sizes over time.

B. Triangle mesh graph, saturated velocities, noise, and formation reconfiguration

We consider $N = 15$ and a triangle mesh graph, built based on proximity in the desired shape. To model non-ideal motions, we consider the speeds ($||\mathbf{u}_i||$) are saturated (with a different maximum speed for each robot), and Gaussian noise is added to every $\mathbf{u}_i$. A gain factor k_u (see Remark 1) is also applied by every robot before saturation. Note that gradient-based coordination controllers (such as ours) are particularly well-suited to withstand perturbations, as noted in [16], [18], [27]; an intuition is that if the angle between robot i's actual (perturbed) motion vector and the negative gradient vector (11) is in $(-\pi/2, \pi/2)$ rad, their inner product is positive and thus i's motion is still reducing γ_d.

Moreover this example features a local reconfiguration of the formation mid-execution. Concretely, at time $t = 15\,\mathrm{s}$, *both* the desired geometry and the formation graph are changed for a subset of robots. A motivation for such a change may be, for instance, reducing the width of the team when passing through a narrow area. As noted in Sect. II, our method uses *local* knowledge of graph structure and desired geometry: this has the advantage that the considered formation reconfiguration can be decided and implemented completely locally, in contrast with Laplacian shape control. Indeed, note that in this example the robots with indices $1, 2, 3, 4, 5, 6, 8, 9$ would not need to participate in the reconfiguration or even be aware of it. Figure 4 illustrates suitable behavior under perturbed motions and formation reconfiguration. Due to the perturbations, $\mathbf{p}(t)$ does not stay constant and $w(t)$ is not guaranteed to be non-increasing.

C. Chained triads graph and unicycle kinematics

Our previous examples were implemented in Matlab. For this one, we ran the simulation in the realistic robotic simulator Gazebo, using the Matlab-ROS interface. This time we used

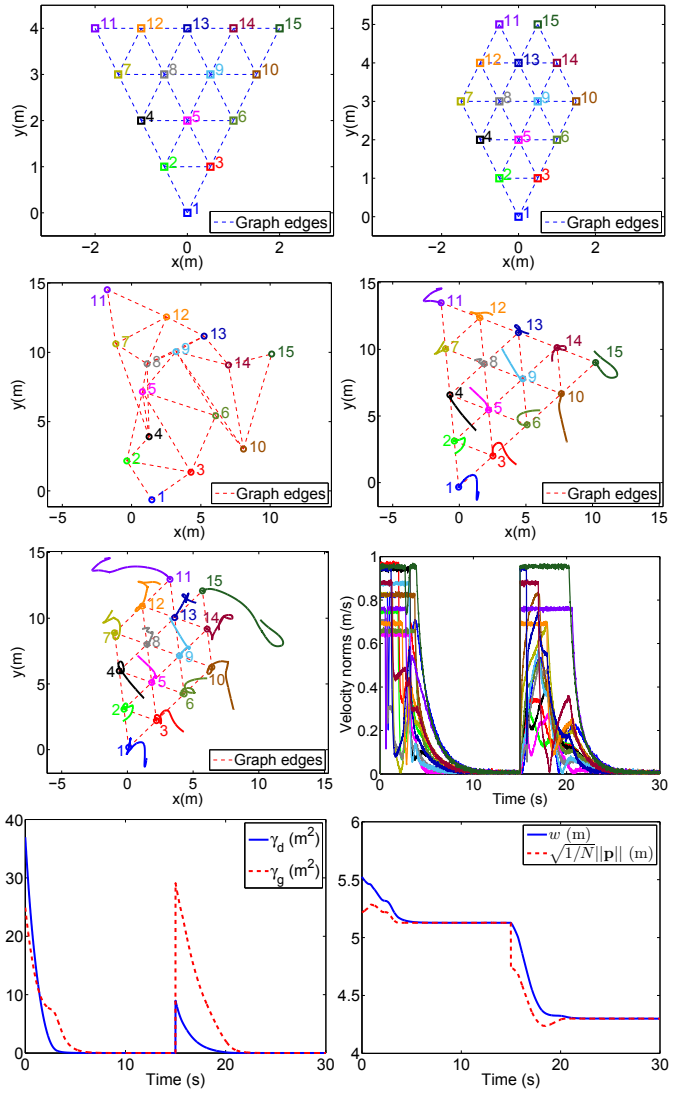

Fig. 4. Results for Sect. V-B. Rows 1-4 (top to bottom) show, from left to right, the following. 1: First desired formation; desired formation after reconfiguration. 2: Initial configuration; robot paths before reconfiguration, converging to the first desired shape. 3: Complete robot paths, converging to the reconfigured desired shape; norms of velocities ($||\mathbf{u}_i||$) over time. 4: Cost functions over time; current (w) and destination sizes over time.

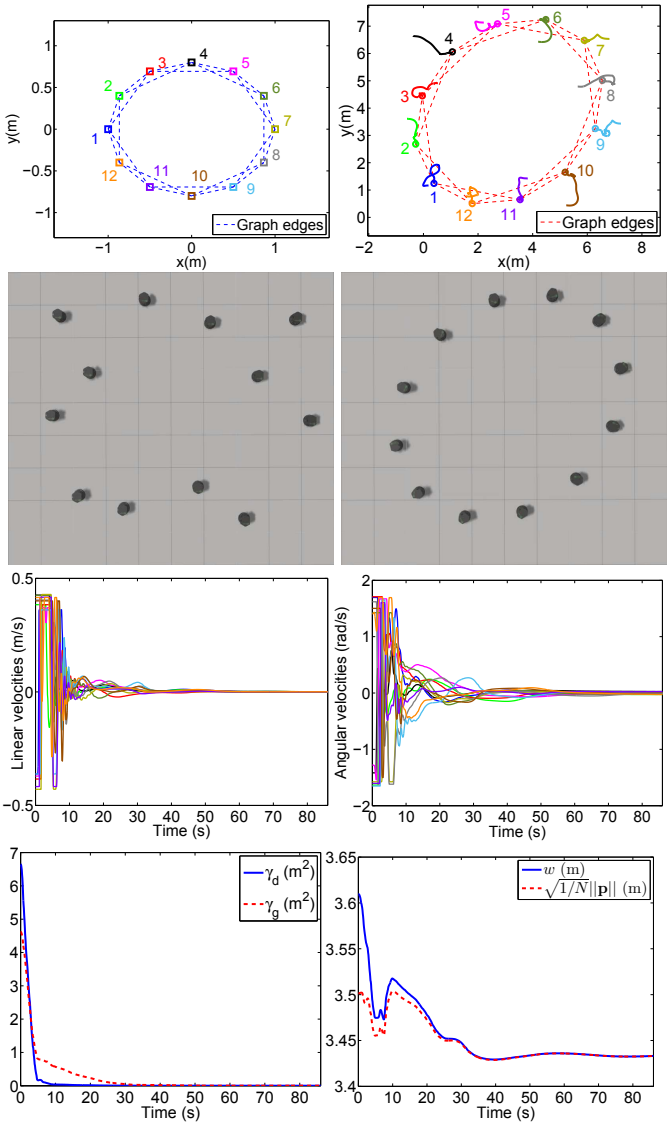

Fig. 5. Results for Sect. V-C. Rows 1-4 (top to bottom) show, from left to right, the following. 1: Desired shape; robot paths, converging to the desired shape. 2: Initial configuration in Gazebo environment; final configuration in Gazebo environment. 3: Linear velocities over time; angular velocities over time. 4: Cost functions over time; current (w) and destination sizes over time.

$N = 12$, the desired geometry was an ellipse and the graph was ring-like, formed by N chained triads: $\mathcal{T}_i = \{i, i+1, i+2\}$ with the indices taken $\bmod N$. We used TurtleBots, which have unicycle kinematics. We controlled them with the strategy in [27, eq. 12]. This strategy takes a generic single-integrator gradient-based multirobot control law (such as ours) and adapts it to unicycles. We call, for each robot i, its linear velocity $v_i \in \mathbb{R}$ and its counterclockwise angular velocity $\omega_i \in \mathbb{R}$, and define $\mathbf{h}_i \in \mathbb{R}^2$ as the unit vector (expressed in the same reference as $\mathbf{u}_i$) along the robot's heading. The control law we use is then:

$$v_i = sat(k_v \mathbf{h}_i' \mathbf{u}_i), \quad \omega_i = sat(k_\omega (\mathbf{S}\mathbf{h}_i)' \mathbf{u}_i), \qquad (27)$$

where $k_v > 0$ and $k_\omega > 0$ are control gains and sat denotes that saturation is applied (i.e., maximum bounds for v_i and ω_i, in absolute values, are imposed). Illustrative plots appear in Fig. 5. Due to the unicycle kinematics, $\mathbf{p}(t)$ is not constant and $w(t)$ increases in certain time intervals. The performance is good, showing the applicability of our approach in a realistic scenario with common kinematic constraints.

D. Tests with sensing noise

Finally, we evaluate the performance under additive Gaussian sensing noise, applied on the robots' measured relative positions. We illustrate the example of Sect. V-A, and a large team ($N = 100$). The results are shown in Fig. 6. As noise affects the destination configuration $\mathbf{p}(t)$ randomly, there is no inherent tendency for $||\mathbf{p}(t)||$ (and, therefore, for the final value of $w(t)$) to drift in a specific direction (downward or upward). Notice that the current sizes w converge to the destination sizes. The controller can deal with noise, and the observed effect is some error, comparable to the noise level, in the final

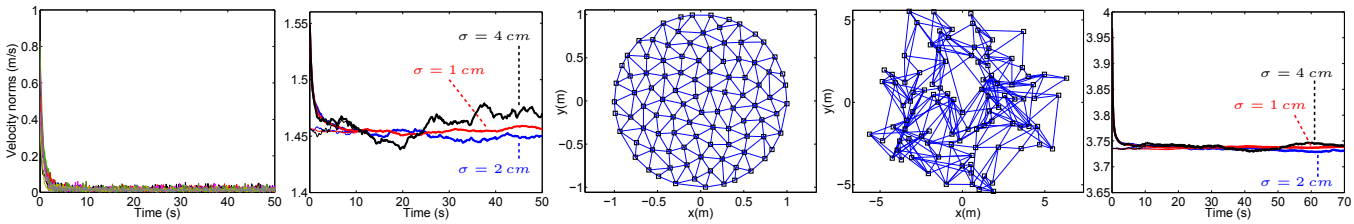

Fig. 6. Tests with sensing noise (Sect. V-D) of standard deviation σ. Left to right: velocity norms with $\sigma = 2\,cm$; and w (thick lines) and destination size $\sqrt{1/N}||\mathbf{p}||$ (thin lines) under different σ, for the example of Fig. 3 (Sect. V-A); desired, and initial, configurations of the 100-robot team with graph edges shown as lines; w (thick lines) and $\sqrt{1/N}||\mathbf{p}||$ (thin lines) under different σ for the 100-robot example. The size magnitudes are shown in meters.

shape. We observed that the average steady-state values of γ_d and γ_g, normalized by N, were of the same order of magnitude as the variance of noise σ^2.

VI. Conclusion

We presented a distributed strategy for multirobot shape control that creates efficient motions and provides advantages in terms of system design and analysis. We also explored connections between geometric and linear algebraic perspectives on the considered problem that can stimulate further developments. To conclude the letter, we next put forward several possible directions in which this work can be improved or extended. First, avoiding collisions is important for practical application; existing reactive approaches such as [28] could be used for this purpose, and we find the study of a collision avoidance mechanism specific to our controller an interesting possibility. The consideration of robots with higher-order dynamics is important in practice. Formation reconfiguration, i.e., allowing the formation graph, formation geometry, or robot labels to change in order to adapt to robot constraints or environment conditions, is a relevant problem as well.

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, March 2015.

[2] L. Krick, M. E. Broucke, and B. A. Francis, "Stabilisation of infinitesimally rigid formations of multi-robot networks," *International Journal of Control*, vol. 82, no. 3, pp. 423–439, March 2009.

[3] H. García de Marina, M. Cao, and B. Jayawardhana, "Controlling rigid formations of mobile agents under inconsistent measurements," *IEEE Transactions on Robotics*, vol. 31, no. 1, pp. 31–39, Feb. 2015.

[4] X. Chen, M. Belabbas, and T. Başar, "Global stabilization of triangulated formations," *SIAM Journal on Control and Optimization*, vol. 55, no. 1, pp. 172–199, 2017.

[5] D. V. Dimarogonas and K. J. Kyriakopoulos, "A connection between formation infeasibility and velocity alignment in kinematic multi-agent systems," *Automatica*, vol. 44, no. 10, pp. 2648–2654, Oct. 2008.

[6] A. N. Bishop, M. Deghat, B. D. O. Anderson, and Y. Hong, "Distributed formation control with relaxed motion requirements," *Int. Journ. Robust and Nonli. Contr.*, vol. 25, no. 17, pp. 3210–3230, Nov. 2015.

[7] M. Aranda, G. López-Nicolás, C. Sagüés, and M. M. Zavlanos, "Distributed formation stabilization using relative position measurements in local coordinates," *IEEE Transactions on Automatic Control*, vol. 61, no. 12, pp. 3925–3935, Dec. 2016.

[8] A. Franchi, C. Masone, V. Grabe, M. Ryll, H. H. Bülthoff, and P. R. Giordano, "Modeling and control of UAV bearing formations with bilateral high-level steering," *The International Journal of Robotics Research*, vol. 31, no. 12, pp. 1504–1525, Oct. 2012.

[9] S. Zhao and D. Zelazo, "Bearing rigidity and almost global bearing-only formation stabilization," *IEEE Transactions on Automatic Control*, vol. 61, no. 5, pp. 1255–1268, May 2016.

[10] R. Tron, J. Thomas, G. Loianno, K. Daniilidis, and V. Kumar, "Bearing-only formation control with auxiliary distance measurements, leaders, and collision avoidance," in *IEEE Conference on Decision and Control (CDC)*, 2016, pp. 1806–1813.

[11] C. J. Stamouli, C. P. Bechlioulis, and K. J. Kyriakopoulos, "Multi-agent formation control based on distributed estimation with prescribed performance," *IEEE Robotics and Automation Letters*, vol. 5, no. 2, pp. 2929–2934, April 2020.

[12] T. Ibuki, S. Wilson, J. Yamauchi, M. Fujita, and M. Egerstedt, "Optimization-based distributed flocking control for multiple rigid bodies," *IEEE Rob. Aut. Letters*, vol. 5, no. 2, pp. 1891–1898, April 2020.

[13] Z. Lin, L. Wang, Z. Han, and M. Fu, "Distributed formation control of multi-agent systems using complex Laplacian," *IEEE Transactions on Automatic Control*, vol. 59, no. 7, pp. 1765–1777, July 2014.

[14] Z. Han, L. Wang, Z. Lin, and R. Zheng, "Formation control with size scaling via a complex Laplacian-based approach," *IEEE Transactions on Cybernetics*, vol. 46, no. 10, pp. 2348–2359, Oct. 2016.

[15] Z. Lin, L. Wang, Z. Han, and M. Fu, "A graph Laplacian approach to coordinate-free formation stabilization for directed networks," *IEEE Trans. Automatic Control*, vol. 61, no. 5, pp. 1269–1280, May 2016.

[16] K. Fathian, T. H. Summers, and N. R. Gans, "Robust distributed formation control of agents with higher-order dynamics," *IEEE Control Systems Letters*, vol. 2, no. 3, pp. 495–500, July 2018.

[17] K. Fathian, D. I. Rachinskii, M. W. Spong, T. H. Summers, and N. R. Gans, "Distributed formation control via mixed barycentric coordinate and distance-based approach," in *American Control Conference (ACC)*, 2019, pp. 51–58.

[18] K. Fathian, S. Safaoui, T. H. Summers, and N. R. Gans, "Robust distributed planar formation control for higher order holonomic and nonholonomic agents," *IEEE Transactions on Robotics*, vol. 37, no. 1, pp. 185–205, Feb. 2021.

[19] J. Guo, Z. Lin, M. Cao, and G. Yan, "Adaptive control schemes for mobile robot formations with triangularised structures," *IET Control Theory & Applications*, vol. 4, no. 9, pp. 1817–1827, Sept. 2010.

[20] M. Basiri, A. N. Bishop, and P. Jensfelt, "Distributed control of triangular formations with angle-only constraints," *Systems & Control Letters*, vol. 59, no. 2, pp. 147–154, Feb. 2010.

[21] L. Guerrero-Bonilla, D. Saldaña, and V. Kumar, "Design guarantees for resilient robot formations on lattices," *IEEE Robotics and Automation Letters*, vol. 4, no. 1, pp. 89–96, Jan. 2019.

[22] S. Zhao and D. Zelazo, "Bearing rigidity theory and its applications for control and estimation of network systems: Life beyond distance rigidity," *IEEE Control Systems Magazine*, vol. 39, no. 2, pp. 66–83, April 2019.

[23] M. Aranda, G. López-Nicolás, C. Sagüés, and M. M. Zavlanos, "Coordinate-free formation stabilization based on relative position measurements," *Automatica*, vol. 57, pp. 11–20, July 2015.

[24] M. Aranda, J. A. Corrales, and Y. Mezouar, "Deformation-based shape control with a multirobot system," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2019, pp. 2174–2180.

[25] D. G. Kendall, "A survey of the statistical theory of shape," *Statist. Sci.*, vol. 4, no. 2, pp. 87–99, May 1989.

[26] S. Umeyama, "Least-squares estimation of transformation parameters between two point patterns," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 13, no. 4, pp. 376–380, April 1991.

[27] S. Zhao, D. V. Dimarogonas, Z. Sun, and D. Bauso, "A general approach to coordination control of mobile agents with motion constraints," *IEEE Trans. Aut. Contr.*, vol. 63, no. 5, pp. 1509–1516, May 2018.

[28] L. Wang, A. D. Ames, and M. Egerstedt, "Safety barrier certificates for collisions-free multirobot systems," *IEEE Transactions on Robotics*, vol. 33, no. 3, pp. 661–674, June 2017.