Marcos-Saavedra, R., Aranda, M., & López-Nicolás, G. (2025). Multirobot control with deformation modes for transport of deformable objects. International Journal of Systems Science, 1–17. https://doi.org/10.1080/00207721.2025.2559138

Multirobot control with deformation modes for transport of deformable objects

Raquel Marcos-Saavedra, Miguel Aranda, and Gonzalo López-Nicolás

***Abstract*—In this article, we present a novel formation control approach for transporting deformable objects with a team of robots. The goal of the proposed system is to reach a target configuration consisting of a desired shape, scale, position and orientation, while allowing linear and quadratic deformations of the robotic formation during the transport. The use of this range of deformation modes enables preserving the integrity of the object while making the transport system highly flexible. Our approach applies a linear combination of terms to regulate the different variables that represent the control goal toward their desired values. We propose controllers both in 2D and 3D workspaces, and provide formal analysis of the exponential convergence of the shape variables and the absence of conflicts between the used control terms. In addition, we illustrate that the presented approach accommodates the use of robots with diverse dynamic models. We validate the proposed controllers via numerical simulations in different settings and experiments with physical unicycle-type robots.**

***Index Terms*—Multi-robot systems; Cooperative transport; Formation control; Manipulation of deformable objects; Deformation modes; Mobile robots.**

I. Introduction

Transporting large or heavy objects using multiple robots can have relevant applications in several domains such as manufacturing, construction, warehouse logistics, or search-and-rescue operations [1]. While most of the existing solutions in this domain have assumed the object to be rigid [2], [3], recent approaches have also addressed the transport of deformable objects. The manipulation of deformable objects by robotic systems is a growing area of study [4]. The particular task of transporting a deformable object with multiple robots [5] is challenging because it requires close coordination, especially when dealing with large, heavy and fragile objects, or if the trajectory to follow needs the team of robots to execute highly specific actions.

Formation control approaches [6]–[8] are interesting in this context as they can allow the robots to cooperatively manipulate deformable objects with accurate motions to prevent damage. In the work [9], controllers for different multirobot formation types were linearly combined, allowing efficient distributed rotation and resizing maneuvers applicable to transport tasks. An approach based on formation control for transporting a flexible load exploiting the robot-object interaction forces was proposed in [10]. In [11]–[13] multiple robots were used to transport a deformable object toward a destination in a coordinated manner considering various kinds of constraints. The work [14] addressed a cooperative human-robot manipulation task with a highly deformable sheet, exploiting force and vision information. In [15], a multiagent manipulation setup with a leader agent was studied, and a position-based dynamics model was used to predict the deformation for different object types. It is also common to use deformable objects, grasped by the robots, to transport a rigid load. For example, [16] proposed a planning approach for a team of wheeled robots transporting a deformable sheet that carried a rigid object, and [17] considered a cable-towed load being transported cooperatively by a group of quadrupedal robots. Teams of aerial vehicles have also been used in prior works to transport, for example, fabrics [18] or cable-suspended loads [19], [20].

A suitable representation of deformation is an important aspect for manipulation systems handling deformable objects. Deformation modes are one of the formal frameworks commonly employed to represent non-rigid behaviors. They have been used, e.g., to model deformable objects in the field of computer graphics. Two examples of works that employed deformation modes in this field are [21], which exploited the Finite Element Method, and [22], which proposed a geometric approach based on shape matching. This latter approach was used in [23] to estimate deformability in robotics applications. Modal analysis has also been recently exploited to control the shape of deformable objects [24].

In this article, we propose a method to transport a deformable object grasped around its contour by multiple robots. We assume the deformation of the transported object is determined by the deformation of the robotic formation, and we focus on the problem of controlling the latter. Our method allows the agents to perform maneuvers that deform the object only with linear and quadratic deformation modes during the trajectory. Some situations require the transported object to be deformed linearly, stretching or shearing it, as considered in [12], [13]. Others, however, also need to deform it in a quadratic way, especially when the task entails changes in direction, causing the object to bend or twist. Our main contribution is a novel control formulation, inspired by [22], that introduces quadratic deformation modes. This enhances adaptability relative to [12], [13] while still maintaining the deformation under control, by restricting it to the described modes. This fact can facilitate preserving the integrity of the transported object. To our knowledge, no prior works have exploited the interesting features of first-order and second-order deformation modes in the problem of transporting deformable objects with multiple robots. Figure 1 illustrates visually our approach.

We formulate a gradient-based strategy for controlling the

The authors are with Instituto Universitario de Investigación en Ingeniería de Aragón (I3A), Universidad de Zaragoza, Spain. Corresponding author: Miguel Aranda (e-mail: miguel.aranda@unizar.es).

shape of the robotic formation. This strategy exploits deformation modes and employs least-squares shape alignment techniques. Additionally, we propose separate control terms for the other formation parameters (scale, position, and orientation), and a full formation controller consisting in a linear combination of all the control terms[1]. We provide formal guarantees for the proposed formation controllers, by studying the convergence of the shape control and the absence of conflicts between the shape control and the control of the other formation parameters. When the robots lie in 2D space, our controller can be formulated analytically. In 3D, computing rotations in the least-squares problems we consider generally relies on numerical procedures [26], [27]. We propose two possible 3D control methods, based on considering infinitesimal rotations and on using the Kabsch algorithm [27], respectively.

The most salient feature of the proposed approach is that, unlike other existing approaches such as the ones in [10], [11], [14]–[16], it can control explicitly the deformation patterns taking place during the transport. The ability to closely control the allowed deformations of a multirobot formation is also our main advantage relative to other formation control works [6]–[9]. Moreover, our approach offers formal guarantees, applicability in both 2D and 3D scenarios, and a computationally simple implementation. In contrast with [12], [13], which already provided multirobot transport approaches based on deformation, here we incorporate second-order deformation modes, which allow for higher flexibility; in addition, we present and formally analyze control laws in 3D.

The proposed approach can be used to transport different types of objects. The most common practical case is the transport of predominantly planar objects: for example, elastic sheets, fabrics, and mattresses. Still, our formulation is general and could also be applied for linear or volumetric objects. Our approach can handle highly deformable objects allowing a flexible transport with controlled deformation, while also being usable in tasks requiring more rigid motions. The considered application scenarios do not include tasks requiring explicit control of the transported object's shape, as our approach cannot handle that type of control. Moreover, it imposes deformation patterns based on standard shape matching, which implicitly assumes a uniform contribution of all parts of the shape. Therefore, it is not a well-suited approach for transporting, e.g., objects with significantly non-uniform material properties. We illustrate the features and capabilities of our approach with diverse simulations and robotic experiments.

II. Notation and Preliminary Definitions

In this article, we assume standard definitions and conventions unless otherwise specified. $\mathbb{R}$, $\mathbb{R}^n$ and $\mathbb{R}^{m\times n}$ denote the set of real numbers, real n-dimensional column vectors, and real matrices with m rows and n columns, respectively. $\mathbf{I}_n \in \mathbb{R}^{n\times n}$ denotes the $n \times n$ identity matrix, $\mathbf{1}_n \in \mathbb{R}^n$ is a column vector of n ones, and $\otimes$ denotes the Kronecker product. We generally do not notate the dependence on time, for the sake of compactness. We will consider a set of N robots, and denote $\mathbf{K} = \mathbf{I}_N - \frac{1}{N}\mathbf{1}_N\mathbf{1}_N^\intercal \in \mathbb{R}^{N\times N}$, which is a centering matrix, and $\mathbf{K}_l = \mathbf{K} \otimes \mathbf{I}_l \in \mathbb{R}^{l\cdot N\times l\cdot N}$. The matrices $\mathbf{K}$ and $\mathbf{K}_l$ are idempotent and symmetric.

For a given matrix $\mathbf{A} \in \mathbb{R}^{m\times n}$, $\mathbf{A}^+ \in \mathbb{R}^{n\times m}$ denotes its Moore-Penrose inverse. Two properties that we use related to this matrix are that $\mathbf{A}^+ = (\mathbf{A}^\intercal\mathbf{A})^+\mathbf{A}^\intercal$ and that $\mathbf{A}\mathbf{A}^+$ is an idempotent and symmetric matrix that projects orthogonally onto the column space, or range, of $\mathbf{A}$ [28, ch. III]. $col_i(\mathbf{A}) \in \mathbb{R}^m$ denotes the $i-th$ column of matrix $\mathbf{A}$, $i \in \{1, 2, \ldots, n\}$, $vec(\mathbf{A}) \in \mathbb{R}^{m\cdot n}$ is a vector resulting from stacking its columns, i.e., $vec(\mathbf{A}) = [col_1(\mathbf{A})^\intercal, \ldots, col_n(\mathbf{A})^\intercal]^\intercal$, and $\|\mathbf{A}\|_F$ denotes its Frobenius norm; if $m = n$, $det(\mathbf{A})$ denotes its determinant and $tr(\mathbf{A})$ denotes its trace. For a given vector $\mathbf{a} \in \mathbb{R}^n$, $\|\mathbf{a}\|$ denotes its Euclidean norm.

[1]Initial steps of this work were previously presented as an extended abstract [25]. Here, we present the proposed control approach in full detail, extend it to 3D, provide formal analysis, and include further experimental validations.

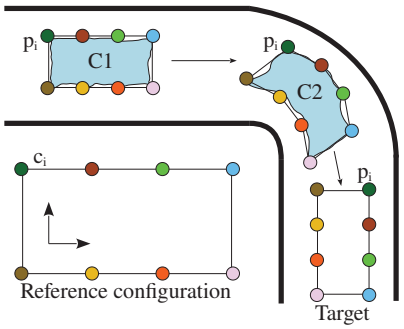


Fig. 1. Illustration of the proposed transport system with eight robots, shown as circles. The system uses a reference configuration and its goal is to reach a target configuration, defined as equal to the reference one up to a desired translation, rotation and scaling. If the task does not require that the object's shape is modified, the formation will be kept in a shape-preserving configuration (C1). However, if the formation needs to deform the object, the deformed formation will be constrained to linear and quadratic deformation modes allowing stretching, shearing, bending and twisting while avoiding other unsuitable and unpredictable deformation patterns (C2).

III. Problem Description

Consider a formation of N robots in 2D or 3D space, grasping through rotational joints an object to be transported as illustrated in Fig. 1. This setup implies that the object can rotate around the point being grasped by each robot, which facilitates the manipulation and transport task. The robot positions in Euclidean space are expressed in a fixed Cartesian coordinate frame. Defining $\mathcal{N} = \{1, 2, \ldots, N\}$, the position of robot $i \in \mathcal{N}$ is denoted by $\mathbf{p}_i \in \mathbb{R}^q$, where $q = 2$ or 3 denotes the number of dimensions. The robots obey single-integrator dynamics, i.e., $\dot{\mathbf{p}}_i = \mathbf{u}_i$, with $\mathbf{u}_i$ being the control input. We define for the full team the stack vectors $\mathbf{p} = [\mathbf{p}_1^\intercal, \ldots, \mathbf{p}_N^\intercal]^\intercal \in \mathbb{R}^{q\cdot N}$ and $\mathbf{u} = [\mathbf{u}_1^\intercal, \ldots, \mathbf{u}_N^\intercal]^\intercal \in \mathbb{R}^{q\cdot N}$. We call a tuple of N ordered positions in $\mathbb{R}^q$ a *configuration*. This tuple, notated $(\mathbf{p}_i)_{i=1}^N$, for the positions $\mathbf{p}_i$ is the *current configuration* of the robotic formation. The *reference configuration* of our approach, $(\mathbf{c}_i)_{i=1}^N$, is defined by a distinct position $\mathbf{c}_i \in \mathbb{R}^q$ associated with every robot i, with stack vector $\mathbf{c} = [\mathbf{c}_1^\intercal, \ldots, \mathbf{c}_N^\intercal]^\intercal \in \mathbb{R}^{q\cdot N}$. We define the reference configuration as constant.

The *target configuration* (Fig. 1) of our approach is equal to the reference one up to translation, rotation and scaling.

In this article, scaling is always considered to be uniform (or isotropic) and by a positive factor. Moreover, rotations are proper, i.e., reflections are not considered. Then, the target configuration is parameterized by: (i) the reference positions $\mathbf{c}_i\ \forall i \in \mathcal{N}$, which encapsulate the desired shape; (ii) a desired scaling, expressed as a positive factor $s_d \in \mathbb{R}$; (iii) a desired position of the formation centroid, $\mathbf{g}_d \in \mathbb{R}^q$; and (iv) a desired orientation, expressed as a counterclockwise rotation of the reference positions by a desired angle $\theta_d \in (-\pi, \pi]$ if $q = 2$ or a desired rotation matrix $\mathbf{R}_d \in SO(3)$ if $q = 3$. The problem we address consists in defining a control law $\mathbf{u}$ to drive the team to the target configuration, while maintaining the deformation under control.

As in [13], we assume the grasping points on the object are suitably placed for enabling its transport, and one can control the deformation of the object and maintain its integrity by properly controlling the shape and scale of the team of robots. This assumption is valid, e.g., for highly deformable objects whose shape adapts to the shape of the team. The assumption also allows for rigid transport, i.e., translation and rotation with tightly maintained shape and scale.

IV. Control with deformation modes in 2D

Suppose the robots move in 2D space, i.e., $q = 2$. We then denote the robot positions as $\mathbf{p}_i = [p_{ix}, p_{iy}]^\intercal \in \mathbb{R}^2$ and the positions in the reference configuration as $\mathbf{c}_i = [c_{ix}, c_{iy}]^\intercal \in \mathbb{R}^2$, $\forall i \in \mathcal{N}$. The control law we propose is a linear combination of separate terms to reach the desired shape, scale, position and orientation, defined as follows:

$$\mathbf{u} = \mathbf{u}_H + \mathbf{u}_G + \mathbf{u}_s + \mathbf{u}_g + \mathbf{u}_\theta. \tag{1}$$

Note that we will use, in every separate control term $\mathbf{u}_a$, a control gain $k_a \in \mathbb{R}$ defined as a positive constant parameter. Next, we present the different terms of this control strategy.

A. Shape control

In order to control the shape of the formation during the transport, two types of configuration are considered: a shape-preserving configuration, and a configuration expressed by deformation modes. Our strategy is to propose two control terms, $\mathbf{u}_H$ and $\mathbf{u}_G$, to move the formation toward these configurations, as explained next.

1) Shape-preserving configuration: This is a configuration in which the positions of the robots, $\mathbf{p}_i$, result from applying translation, rotation and scaling to the reference positions, $\mathbf{c}_i$. The configuration labeled C1 in Fig. 1 is an example of this type of configuration. Consider a matrix and a vector of the form

$$\mathbf{H} = \begin{bmatrix} h_1 & -h_2 \\ h_2 & h_1 \end{bmatrix} \in \mathbb{R}^{2\times 2}, \quad \mathbf{t} = \begin{bmatrix} t_x \\ t_y \end{bmatrix} \in \mathbb{R}^2, \tag{2}$$

where h_1 and h_2 are not both zero but otherwise have arbitrary values, and t_x and t_y have arbitrary values. Then, consider the following condition for a configuration:

$$\mathbf{p}_i = \mathbf{H}\mathbf{c}_i + \mathbf{t} \quad \forall i \in \mathcal{N}. \tag{3}$$

Note that $\mathbf{H}$ encodes a generic rotation and a generic scaling, and $\mathbf{t}$ a generic translation, being applied to all the reference positions. Therefore, (3) implies that the formation is in a shape-preserving configuration. For $N > 2$, the current configuration of the team $(\mathbf{p}_i)_{i=1}^N$ may not be shape-preserving; this will be the case when there are no $\mathbf{H}$, $\mathbf{t}$ such that (3) holds. Our strategy for making the team achieve the condition (3) is to define a specific shape-preserving configuration $(\mathbf{p}_{H,i})_{i=1}^N$ in $\mathbb{R}^2$ of the form

$$\mathbf{p}_{H,i} = \mathbf{H}(\mathbf{p}, \mathbf{c})\, \mathbf{c}_i + \mathbf{t}(\mathbf{p}, \mathbf{c}) \quad \forall i \in \mathcal{N}, \tag{4}$$

and to propose a controller that continuously moves every $\mathbf{p}_i$ toward $\mathbf{p}_{H,i}$, in order to satisfy the condition

$$\mathbf{p}_i = \mathbf{p}_{H,i} \quad \forall i \in \mathcal{N}. \tag{5}$$

Notice that this condition implies (3). The notations $\mathbf{H}(\mathbf{p}, \mathbf{c})$ and $\mathbf{t}(\mathbf{p}, \mathbf{c})$ in (4) mean that $\mathbf{H}$ and $\mathbf{t}$ are chosen as a function of $\mathbf{p}$ and $\mathbf{c}$. Specifically, we choose $\mathbf{H}$ and $\mathbf{t}$ based on least-squares alignment between the two configurations encoded by $\mathbf{p}$ and $\mathbf{c}$, respectively. The procedure is explained next.

We choose the translation $\mathbf{t}$ so that $(\mathbf{p}_{H,i})_{i=1}^N$ has the same centroid as $(\mathbf{p}_i)_{i=1}^N$. This choice has been shown to be optimal in other related problems [27], and it will facilitate controlling the formation centroid with a separate specific control term. Concretely, we first define the centroids

$$\bar{\mathbf{p}} = \frac{1}{N}\sum_{i=1}^{N} \mathbf{p}_i, \quad \bar{\mathbf{c}} = \frac{1}{N}\sum_{i=1}^{N} \mathbf{c}_i. \tag{6}$$

Then, we choose $\mathbf{t} = \bar{\mathbf{p}} - \mathbf{H}\bar{\mathbf{c}}$. Therefore, (4) becomes

$$\mathbf{p}_{H,i} = \bar{\mathbf{p}} + \mathbf{H}(\mathbf{c}_i - \bar{\mathbf{c}}) \quad \forall i \in \mathcal{N}. \tag{7}$$

Notice, then, that the condition (5) can be given as

$$\mathbf{p}_i - \bar{\mathbf{p}} = \mathbf{H}(\mathbf{c}_i - \bar{\mathbf{c}}) \quad \forall i \in \mathcal{N}. \tag{8}$$

We will now express (8) for the complete formation in a compact manner using the centering matrix $\mathbf{K}$. Defining

$$\mathbf{P} = \begin{bmatrix} p_{1x} & \cdots & p_{Nx} \\ p_{1y} & \cdots & p_{Ny} \end{bmatrix}, \ \mathbf{C} = \begin{bmatrix} c_{1x} & \cdots & c_{Nx} \\ c_{1y} & \cdots & c_{Ny} \end{bmatrix}, \tag{9}$$

we have that $\mathbf{PK} = \mathbf{HCK}$. Let us define $\mathbf{S} = [[0,1]^\intercal, [-1,0]^\intercal]$. Notice that $\mathbf{H} = h_1\mathbf{I}_2 + h_2\mathbf{S}$ and hence

$$\mathbf{PK} = (h_1\mathbf{C} + h_2\mathbf{SC}) \cdot \mathbf{K}. \tag{10}$$

Next, we express (10) in a more convenient vectorized form. Note that $vec(\mathbf{P}) = \mathbf{p}$ and $vec(\mathbf{C}) = \mathbf{c}$. Let us define $\mathbf{c}_\perp = vec(\mathbf{SC}) = [-c_{1y}, c_{1x}, \ldots, -c_{Ny}, c_{Nx}]^\intercal \in \mathbb{R}^{2N}$. We apply vectorization on both sides of (10), obtaining[2]

$$\mathbf{K}_2\mathbf{p} = \mathbf{K}_2(h_1\mathbf{c} + h_2\mathbf{c}_\perp) = \mathbf{K}_2 \begin{bmatrix} \mathbf{c} & \mathbf{c}_\perp \end{bmatrix} \begin{bmatrix} h_1 \\ h_2 \end{bmatrix}. \tag{11}$$

We now define the $2N \times 2$ matrix

$$\mathbf{C}_H = \mathbf{K}_2 \begin{bmatrix} \mathbf{c} & \mathbf{c}_\perp \end{bmatrix} = \mathbf{K}_2 \begin{bmatrix} c_{1x} & c_{1y} & \cdots & c_{Nx} & c_{Ny} \\ -c_{1y} & c_{1x} & \cdots & -c_{Ny} & c_{Nx} \end{bmatrix}^\intercal, \tag{12}$$

and the vector $\mathbf{h}_H = [h_1, h_2]^\intercal \in \mathbb{R}^2$. Then, from (11), the shape preservation condition (5) takes the final form

$$\mathbf{K}_2\mathbf{p} = \mathbf{C}_H\mathbf{h}_H. \tag{13}$$

[2]We use here that for any two matrices $\mathbf{A}$, $\mathbf{B}$ of compatible dimensions, $vec(\mathbf{AB}) = (\mathbf{B}^\intercal \otimes \mathbf{I}_k)vec(\mathbf{A})$ where k is the number of rows of $\mathbf{A}$.

Based on this expression, we now choose the values in $\mathbf{h}_H$ so that $(\mathbf{p}_{H,i})_{i=1}^N$ is closest to $(\mathbf{p}_i)_{i=1}^N$ in a least-squares sense, as discussed above. In particular, we choose $\mathbf{h}_H$ so that, for the current $\mathbf{p}$, $\|\mathbf{K}_2\mathbf{p} - \mathbf{C}_H\mathbf{h}_H\|$ is minimum. This is a linear least-squares problem, for which further details are given in the Appendix. Hence, from this point onward, we choose

$$\mathbf{h}_H = \mathbf{C}_H^+\mathbf{K}_2\mathbf{p} = \mathbf{C}_H^+\mathbf{p}, \tag{14}$$

where the second equality results from the fact[3] that $\mathbf{C}_H^+\mathbf{K}_2 = \mathbf{C}_H^+$. Note that it is theoretically possible to have $\mathbf{h}_H = [0,0]^\intercal$ in (14). In such a case, $(\mathbf{p}_{H,i})_{i=1}^N$ is a configuration where all positions are equal. This is a particular degenerate case, corresponding to zero-measure team configurations with very large deformations. This case is not problematic in practice and we disregard it, as in prior works [13]. Let us define

$$\mathbf{A}_H = \mathbf{K}_2 - \mathbf{C}_H\mathbf{C}_H^+. \tag{15}$$

$\mathbf{A}_H$ is a constant matrix that has the relevant properties of being idempotent, symmetric and positive semidefinite. We formulate the following cost function, which is associated with shape preservation:

$$\gamma_H = \frac{1}{2}\|\mathbf{K}_2\mathbf{p} - \mathbf{C}_H\mathbf{h}_H\|^2 = \frac{1}{2}\mathbf{p}^\intercal\mathbf{A}_H\mathbf{p}. \tag{16}$$

Then, we propose a controller for preserving the shape of the team, by moving in the direction of the negative gradient of $\gamma_H(\mathbf{p})$, as

$$\mathbf{u}_H = -k_H\mathbf{A}_H\mathbf{p}. \tag{17}$$

For every robot i, implementing this control simply requires multiplying the coefficients in the two rows of the constant matrix $\mathbf{A}_H$ corresponding to index i by the robot positions gathered in $\mathbf{p}$. This valuable simplicity is due to the fact that the parameters in $\mathbf{h}_H$ (equivalently, in $\mathbf{H}$) from (14) depend linearly on the positions $(\mathbf{p}_i)_{i=1}^N$. It is instructive to also interpret the control for every robot i by noticing that $\mathbf{u}_H = -k_H(\mathbf{K}_2\mathbf{p} - \mathbf{C}_H\mathbf{h}_H)$, which can be related back to (13) and, in turn, to the separate expressions for every i in (8). In this way, calling $\mathbf{u}_H = [\mathbf{u}_{H,1}^\intercal, \ldots, \mathbf{u}_{H,N}^\intercal]^\intercal$, the control for every robot $i \in \mathcal{N}$ can be expressed as

$$\mathbf{u}_{H,i} = k_H\big(\mathbf{H}(\mathbf{c}_i - \bar{\mathbf{c}}) + \bar{\mathbf{p}} - \mathbf{p}_i\big) = k_H(\mathbf{p}_{H,i} - \mathbf{p}_i), \tag{18}$$

which is consistent with the goal stated above of driving every $\mathbf{p}_i$ toward $\mathbf{p}_{H,i}$.

2) Configuration expressed by deformation modes: The second configuration we consider is one that can be expressed by deformation modes up to order two, which will allow the formation to deform in a controlled way. The configuration labeled C2 in Fig. 1 is an example of this type of configuration. Our formulation for the deformation modes is inspired by [22]. Note that this formulation, which we present next, follows closely the developments used in Section IV-A1 for the shape-preserving configuration. We use

[3] $\mathbf{C}_H^+\mathbf{K}_2 = (\mathbf{C}_H^\intercal\mathbf{C}_H)^+\mathbf{C}_H^\intercal\cdot\mathbf{K}_2$, and here $\mathbf{C}_H^\intercal\mathbf{K}_2 = (\mathbf{K}_2\mathbf{C}_H)^\intercal = \mathbf{C}_H^\intercal$, which follows from $\mathbf{K}_2$ being symmetric and from $\mathbf{K}_2\mathbf{C}_H = \mathbf{C}_H$ due to $\mathbf{K}_2$ being idempotent. Hence, $\mathbf{C}_H^+\mathbf{K}_2 = (\mathbf{C}_H^\intercal\mathbf{C}_H)^+\mathbf{C}_H^\intercal = \mathbf{C}_H^+$.

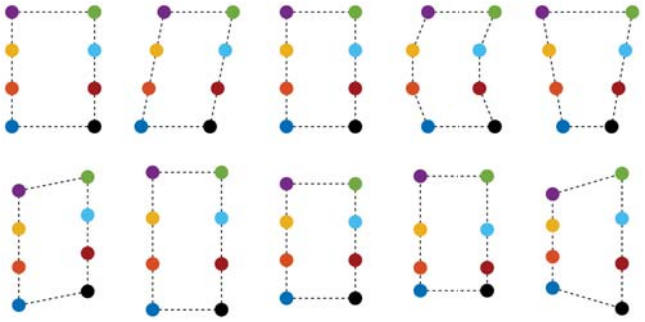


Fig. 2. Representation of deformation modes for a rectangular reference configuration with $N = 8$. Each shape (j,k) in the 2×5 panel is obtained as $\mathbf{c}_i + \mathbf{G}\,col_i(\mathbf{D})\ \forall i$, where $\mathbf{G}$, having the form in (19), has all zeros except for $\mathbf{G}_{j,k} = \alpha$, with α a positive number. Figure inspired by Fig. 5 in [22].

a similar condition to the one given above: now, the condition is $\mathbf{p}_i = \mathbf{G}\cdot col_i(\mathbf{D}) + \mathbf{t}_G$ for every robot i, where we define

$$\mathbf{G} = \begin{bmatrix} l_1 & l_2 & q_1 & q_2 & m_1 \\ l_3 & l_4 & q_3 & q_4 & m_2 \end{bmatrix},\ \mathbf{D} = \begin{bmatrix} c_{1x} & \cdots & c_{Nx} \\ c_{1y} & \cdots & c_{Ny} \\ c_{1x}^2 & \cdots & c_{Nx}^2 \\ c_{1y}^2 & \cdots & c_{Ny}^2 \\ c_{1x}\,c_{1y} & \cdots & c_{Nx}\,c_{Ny} \end{bmatrix}, \tag{19}$$

and a vector $\mathbf{t}_G \in \mathbb{R}^2$ that represents a translation. The form of matrix $\mathbf{D} \in \mathbb{R}^{5\times N}$ determines the deformation modes that we allow: linear, purely quadratic, and mixed deformations of the reference geometry encoded by $(\mathbf{c}_i)_{i=1}^N$. In the matrix $\mathbf{G} \in \mathbb{R}^{2\times5}$, l_i, q_i and m_i are the parameters associated with these three modes. Linear terms can only represent stretch and shear, whereas purely quadratic and mixed terms can represent bend and twist, thus expanding the range of possible deformations. The shapes displayed in Fig. 2 exemplify the available deformation modes for a rectangular reference configuration.

We define, as done above, a specific configuration $(\mathbf{p}_{G,i})_{i=1}^N$ expressed by deformation modes, i.e.:

$$\mathbf{p}_{G,i} = \mathbf{G}(\mathbf{p},\mathbf{c})\cdot col_i(\mathbf{D}) + \mathbf{t}_G(\mathbf{p},\mathbf{c}) \quad \forall i \in \mathcal{N}, \tag{20}$$

where $\mathbf{t}_G$ is now chosen so that the centroid of $(\mathbf{p}_{G,i})_{i=1}^N$ is the same as the centroid of $(\mathbf{p}_i)_{i=1}^N$:

$$\mathbf{t}_G = \bar{\mathbf{p}} - \mathbf{G}\bar{\mathbf{d}}, \ \text{ with } \ \bar{\mathbf{d}} = \frac{1}{N}\sum_{i=1}^N col_i(\mathbf{D}), \tag{21}$$

and $\mathbf{G}$ will be chosen by using least-squares shape alignment. Notice that we can write a similar expression to the expression (7) for $\mathbf{p}_{H,i}$, as follows:

$$\mathbf{p}_{G,i} = \bar{\mathbf{p}} + \mathbf{G}(col_i(\mathbf{D}) - \bar{\mathbf{d}}) \quad \forall i \in \mathcal{N}. \tag{22}$$

Then, we will propose a controller to make every $\mathbf{p}_i$ move toward $\mathbf{p}_{G,i}$ with the goal of making $\mathbf{p}_i = \mathbf{p}_{G,i}$. This latter condition is then similar to (8) above, since one has

$$\mathbf{p}_i - \bar{\mathbf{p}} = \mathbf{G}(col_i(\mathbf{D}) - \bar{\mathbf{d}}) \quad \forall i \in \mathcal{N}. \tag{23}$$

Therefore, for the full formation, one has in compact form that $\mathbf{PK} = \mathbf{GDK}$. Applying again vectorization, this condition is transformed into $\mathbf{K}_2\mathbf{p} = \mathbf{C}_G\mathbf{h}_G$, where we define

$$\mathbf{C}_G = \mathbf{K}_2 \begin{bmatrix} \mathbf{L}_1 & \mathbf{Q}_1 & \mathbf{M}_1 \\ \vdots & \vdots & \vdots \\ \mathbf{L}_N & \mathbf{Q}_N & \mathbf{M}_N \end{bmatrix} \in \mathbb{R}^{2N\times 10}, \tag{24}$$

$$\mathbf{L}_i = \begin{bmatrix} c_{ix} & c_{iy} & 0 & 0 \\ 0 & 0 & c_{ix} & c_{iy} \end{bmatrix} \in \mathbb{R}^{2\times 4}, \tag{25}$$

$$\mathbf{Q}_i = \begin{bmatrix} c_{ix}^2 & c_{iy}^2 & 0 & 0 \\ 0 & 0 & c_{ix}^2 & c_{iy}^2 \end{bmatrix} \in \mathbb{R}^{2\times 4}, \tag{26}$$

$$\mathbf{M}_i = \begin{bmatrix} c_{ix}c_{iy} & 0 \\ 0 & c_{ix}c_{iy} \end{bmatrix} \in \mathbb{R}^{2\times 2}, \tag{27}$$

and as done previously with the vector $\mathbf{h}_H$, we now define a vector corresponding to the components of $\mathbf{G}$: $\mathbf{h}_G = [l_1, l_2, l_3, l_4, q_1, q_2, q_3, q_4, m_1, m_2]^\intercal \in \mathbb{R}^{10}$. Again, we select $\mathbf{h}_G$ using a least-squares strategy, so that $\|\mathbf{K}_2\mathbf{p} - \mathbf{C}_G\mathbf{h}_G\|$ is minimized for the current $\mathbf{p}$:

$$\mathbf{h}_G = \mathbf{C}_G^+\mathbf{K}_2\mathbf{p} = \mathbf{C}_G^+\mathbf{p}. \tag{28}$$

We now define the matrix

$$\mathbf{A}_G = \mathbf{K}_2 - \mathbf{C}_G\mathbf{C}_G^+, \tag{29}$$

which is constant and, similarly to $\mathbf{A}_H$ defined above, has the useful properties of being idempotent, symmetric and positive semidefinite. Then, we define a cost function associated with the deformation modes as

$$\gamma_G = \frac{1}{2}\|\mathbf{K}_2\mathbf{p} - \mathbf{C}_G\mathbf{h}_G\|^2 = \frac{1}{2}\mathbf{p}^\intercal\mathbf{A}_G\mathbf{p}. \tag{30}$$

We propose a control term, $\mathbf{u}_G$, following the negative gradient of $\gamma_G(\mathbf{p})$:

$$\mathbf{u}_G = -k_G\mathbf{A}_G\mathbf{p}. \tag{31}$$

This term aims to maintain a controlled deformation by continuously moving the team toward the configuration $(\mathbf{p}_{G,i})_{i=1}^N$, in which the deformation is constrained to the allowed modes. The implementation of robot i's control term according to (31) simply requires multiplying the corresponding coefficients in the constant matrix $\mathbf{A}_G$ by the positions $\mathbf{p}_i$ collected in $\mathbf{p}$.

It is important to note that any shape-preserving configuration can be represented by the allowed deformation modes, with $\mathbf{G} = [\mathbf{H}, \mathbf{0}_{2\times 3}]$ in (19). Hence, if the current configuration $(\mathbf{p}_i)_{i=1}^N$ is shape-preserving, then it is already a configuration that can be expressed by the considered deformation modes, and therefore the control term $\mathbf{u}_G$ will not produce any motion. In this way, this term does not induce any new deformation; it just controls the deformation when it exists, by restricting it to the allowed modes. Calling $\mathbf{u}_G = [\mathbf{u}_{G,1}^\intercal, \ldots, \mathbf{u}_{G,N}^\intercal]^\intercal$, this control for every robot $i \in \mathcal{N}$ can be expressed as

$$\mathbf{u}_{G,i} = k_G\big(\mathbf{G}(col_i(\mathbf{D}) - \bar{\mathbf{d}}) + \bar{\mathbf{p}} - \mathbf{p}_i\big) = k_G(\mathbf{p}_{G,i} - \mathbf{p}_i), \tag{32}$$

which corresponds to driving every $\mathbf{p}_i$ toward $\mathbf{p}_{G,i}$.

For low numbers of robots and certain reference geometries (e.g., in certain setups with up to six robots), our control $\mathbf{u}_G$ for maintaining a linear and quadratic deformation does not act (i.e., it is always zero). These are cases where, due to rank conditions, *any* $\mathbf{p}$ can be expressed as a linear and quadratic deformation that makes $\gamma_G = 0$. This implies that $\mathbf{A}_G = \mathbf{0}$. We do not address a specific analysis of these cases. A similar fact occurs with our deformation controller in 3D space, described in Section V. Note that one can control the team to maintain a linear-only deformation, by simply removing all $\mathbf{Q}_i$ and $\mathbf{M}_i$ from $\mathbf{C}_G$ in (24).

Least-squares techniques that are similar to the ones we discussed have been used when defining deformation-related functions on the set of spatial points that discretely represent a deformable object [22], [31]. In this work, as in [12], [13], our idea is to use these techniques to define deformation-related cost functions on the set of the positions of the robots that transport a deformable object.

B. Scale, translation and rotation control

To fully control the deformation, we also control the team's scale using the variable $s = \|\mathbf{h}_H\|$. For achieving the desired scale, s_d, we propose the following control term, where as noted above, $s > 0$ (i.e., $\mathbf{h}_H \neq [0,0]^\intercal$) can be assumed:

$$\mathbf{u}_s = k_s\,(s_d - s)\,(1/s)\,\mathbf{C}_H\mathbf{h}_H. \tag{33}$$

Translation and rotation of the formation do not alter the relative positions of the robots, so they do not affect the object's deformation. The translation controller is responsible for driving the team of robots as a whole to achieve the desired absolute position, $\mathbf{g}_d$, of the formation centroid, defined as $\mathbf{g} = \bar{\mathbf{p}} = \frac{1}{N}\mathbf{P}\,\mathbf{1}_N$. The proposed controller is

$$\mathbf{u}_g = k_g\,\mathbf{1}_N \otimes (\mathbf{g}_d - \mathbf{g}). \tag{34}$$

The rotation controller rotates the shape around the formation centroid until the angle that defines the orientation of the formation, defined as $\theta = \text{atan2}\,(h_2, h_1) \in (-\pi, \pi]$, reaches the desired value, θ_d. Recalling that $\mathbf{S} = [[0,1]^\intercal, [-1,0]^\intercal]$, the rotation controller we propose has the form

$$\mathbf{u}_\theta = k_\theta\,(\theta_d - \theta)\,(\mathbf{I}_N \otimes \mathbf{S})\,\mathbf{C}_H\mathbf{h}_H. \tag{35}$$

For convenience, one can take $\theta_d = 0$ without loss of generality[4].

C. Controller analysis

It is interesting that the two shape control terms, $\mathbf{u}_H$ and $\mathbf{u}_G$, have a simple linear dependence on the robot positions. This fact facilitates their theoretical analysis. Next, we present two formal results for the proposed controller, whose proofs can be found in the Appendix.

Theorem 1. *If $s(t=0) > 0$, under the action of the control $\mathbf{u}_H + \mathbf{u}_G$ from* (17)*,* (31)*, $\mathbf{p}$ converges globally and exponentially fast to a static state for which the shape preservation condition is satisfied; i.e., $\forall i \in \mathcal{N}$, $\mathbf{p}_i = \mathbf{H}\mathbf{c}_i + \mathbf{t}$ for some $\mathbf{t} \in \mathbb{R}^2$ and a non-zero $\mathbf{H} \in \mathbb{R}^{2\times 2}$ having the form in* (2)*. In addition, $\mathbf{g}$ and $\mathbf{h}_H$ are invariant.*

[4]For any $\theta_d \in (-\pi, \pi]$, one can redefine $\mathbf{c}_i = [[\cos(\theta_d), \sin(\theta_d)]^\intercal, [-\sin(\theta_d), \cos(\theta_d)]^\intercal]\,\mathbf{c}_i$ $\forall i \in \mathcal{N}$, and then redefine $\theta_d = 0$.

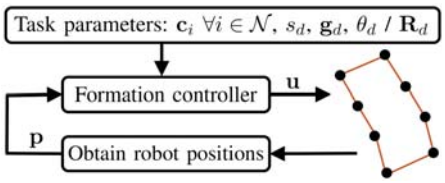


Fig. 3. Diagram showing the main components of our control approach.

Assuming $s(t=0) > 0$ in Theorem 1 is reasonable because, as already noted, this condition is always satisfied except for degenerate, zero-measure initial configurations with very large deformations. Theorem 1 shows that the shape control $\mathbf{u}_H + \mathbf{u}_G$ drives the team to a shape-preserving configuration if it is possible to do so, and keeps it close to a linearly and quadratically deformed one if it is not (e.g., due to the presence of exogenous constraints such as obstacles). The fact that $\mathbf{g}$ and $\mathbf{h}_H$ are invariant is important because it means s, $\mathbf{g}$, and θ are unaffected by $\mathbf{u}_H + \mathbf{u}_G$, which facilitates controlling these variables with the control terms $\mathbf{u}_s$, $\mathbf{u}_g$, and $\mathbf{u}_\theta$ respectively. Conversely, these three control terms do not modify the shape-related cost functions either, as shown next.

Proposition 1. *The cost functions γ_H and γ_G are invariant under the actions of each of the control terms $\mathbf{u}_s$, $\mathbf{u}_g$, and $\mathbf{u}_\theta$.*

This result shows that there are no conflicts between the control of the shape and the control of the other three variables. This relevant fact supports the suitability of the linear combination of control terms proposed in this article. The terms $\mathbf{u}_s$, $\mathbf{u}_g$, and $\mathbf{u}_\theta$ were used in [12] (equations 17, 15, and 16 in [12], respectively), with a different matrix-vector format. Even when the three of them are used simultaneously, these terms allow for decoupled convergences of s, $\mathbf{g}$, and θ toward their desired values. We refer to [12, Prop. 2 and Thm. 1] for further details.

D. Controller implementation

A representation of the main components of the proposed controller is shown in Fig. 3. Note that the controller for every robot uses the information of the current positions of all the robots; i.e., $\mathbf{p}_i\ \forall i \in \mathcal{N}$. The advantage of this is that the use of global information facilitates obtaining the highly coordinated behaviors required for the addressed multirobot transport task. In this particular task, unlike in other types of multirobot tasks, the number of robots is usually moderate and the robots typically stay close to each other; it is then reasonable to assume that every robot can know the positions of the others by using its own sensors or via data exchanges using communications. An alternative is to have a central node that computes and sends the motion commands to the robots. The control law (1) uses elementary closed-form expressions and does not require iterative procedures or optimizations. The matrices involved ($\mathbf{C}_H$, $\mathbf{C}_G$ and their Moore-Penrose inverses) are constant and can be pre-stored. The 3D control extensions presented in the next section use similar types of operations, in addition to a singular value decomposition of a 3×3 matrix. These facts show that the run-time computational requirements of our approach are not high. We provide results in Section VII supporting this claim.

If the proposed approach is used with many robots, sensing and communication demands can grow significantly. It is interesting to note that our formulation can admit a distributed implementation, which is a way to alleviate these issues. A given robot i can compute all its control terms in (1) by using the pre-stored constant parameters of the target configuration, measuring its own time-varying position $\mathbf{p}_i$, and estimating the time-varying quantities in $\mathbf{g}$, $\mathbf{h}_H$, and $\mathbf{h}_G$. As these quantities are linear combinations of the robots' positions, they can be estimated by running communications-based average consensus algorithms; for example, the algorithm structures in [29, eqs. 16a, 16b]. We leave the detailed study of the distributed implementation of our approach for future work.

V. Extension to 3D space

In 3D space (i.e., $q = 3$), there is more freedom of movement and the handling of rotations becomes more complex. It is not possible to use, as we did in 2D, a least-squares scaling and rotation transformation that depends linearly on the robot positions. Generally, computing least-squares 3D rotations requires numerical procedures [26]. We propose two approaches based on infinitesimal rotations and on the Kabsch algorithm, respectively. Note that, for notational simplicity, we generally use the same names for the variables as in 2D.

A. Control with infinitesimal rotations

Inspired by [26], we consider a transformation matrix

$$\mathbf{H} = \begin{bmatrix} h_s & -h_{rz} & h_{ry} \\ h_{rz} & h_s & -h_{rx} \\ -h_{ry} & h_{rx} & h_s \end{bmatrix} \in \mathbb{R}^{3\times3}, \tag{36}$$

with $h_s > 0$. This transformation has the form of a scaling and a rotation by infinitesimal angles. It can approximate a scaling and rotation transformation when the rotations are by small angles [26]. The motivation for using this $\mathbf{H}$ is that it allows us to formulate the controller analytically and following similar principles to the ones of the 2D case.

The shape preservation condition we consider, analogously to the 2D case, is $\mathbf{p}_i = \mathbf{H}\mathbf{c}_i + \mathbf{t}$ with $\mathbf{p}_i = [p_{ix}, p_{iy}, p_{iz}]^\intercal$, $\mathbf{c}_i = [c_{ix}, c_{iy}, c_{iz}]^\intercal$, $\mathbf{t} = [t_x, t_y, t_z]^\intercal \in \mathbb{R}^3$, $\forall i \in \mathcal{N}$. An important difference with respect to the 2D case is that in 3D, this condition does not imply exact shape preservation, because $\mathbf{H}$ is an approximated scaling and rotation transformation. Following the same procedure as in 2D, we define a matrix $\mathbf{C}_H = \mathbf{K}_3[\mathbf{C}_{H,1}, \ldots, \mathbf{C}_{H,N}]^\intercal \in \mathbb{R}^{3N\times4}$, where

$$\mathbf{C}_{H,i} = \begin{bmatrix} c_{ix} & c_{iy} & c_{iz} \\ 0 & -c_{iz} & c_{iy} \\ c_{iz} & 0 & -c_{ix} \\ -c_{iy} & c_{ix} & 0 \end{bmatrix}. \tag{37}$$

Then, we use again a least-squares strategy for shape alignment, which is similar to the one employed in [26], to choose the values of $\mathbf{H}$ via the parameter vector $\mathbf{h}_H = [h_s, \mathbf{h}_r^\intercal]^\intercal \in \mathbb{R}^4$ where $\mathbf{h}_r = [h_{rx}, h_{ry}, h_{rz}]^\intercal$. By applying this strategy, defining $\gamma_H = \frac{1}{2}\|\mathbf{K}_3\mathbf{p} - \mathbf{C}_H\mathbf{h}_H\|^2 = \frac{1}{2}\mathbf{p}^\intercal\mathbf{A}_H\mathbf{p}$ where

$\mathbf{A}_H = \mathbf{K}_3 - \mathbf{C}_H\mathbf{C}_H^+$, we can formulate the control term for preserving the shape of the team as in (17).

Let us define $\mathbf{c}_i^l = [c_{ix}, c_{iy}, c_{iz}]$, $\mathbf{c}_i^q = [c_{ix}^2, c_{iy}^2, c_{iz}^2]$, $\mathbf{c}_i^m = [c_{ix}c_{iy}, c_{iy}c_{iz}, c_{iz}c_{ix}]$. To represent a configuration deformed by modes up to order two, we use the condition that, for every i, it has to be satisfied that $\mathbf{p}_i = \mathbf{G} \cdot [\mathbf{c}_i^l, \mathbf{c}_i^q, \mathbf{c}_i^m]^\intercal + \mathbf{t}_G$, for some $\mathbf{t}_G \in \mathbb{R}^3$ and defining $\mathbf{G}$ as

$$\mathbf{G} = \begin{bmatrix} l_1 & l_2 & l_3 & q_1 & q_2 & q_3 & m_1 & m_2 & m_3 \\ l_4 & l_5 & l_6 & q_4 & q_5 & q_6 & m_4 & m_5 & m_6 \\ l_7 & l_8 & l_9 & q_7 & q_8 & q_9 & m_7 & m_8 & m_9 \end{bmatrix} \in \mathbb{R}^{3\times 9}. \tag{38}$$

We define the matrix $\mathbf{C}_G$ with the linear ($\mathbf{L}_i$), purely quadratic ($\mathbf{Q}_i$) and mixed ($\mathbf{M}_i$) terms as

$$\mathbf{C}_G = \mathbf{K}_3 \cdot \begin{bmatrix} \mathbf{L}_1 & \mathbf{Q}_1 & \mathbf{M}_1 \\ \vdots & \vdots & \vdots \\ \mathbf{L}_N & \mathbf{Q}_N & \mathbf{M}_N \end{bmatrix} \in \mathbb{R}^{3N\times 27}, \tag{39}$$

$$\mathbf{L}_i = \begin{bmatrix} \mathbf{c}_i^l & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{c}_i^l & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{c}_i^l \end{bmatrix} \in \mathbb{R}^{3\times 9}, \tag{40}$$

$$\mathbf{Q}_i = \begin{bmatrix} \mathbf{c}_i^q & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{c}_i^q & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{c}_i^q \end{bmatrix} \in \mathbb{R}^{3\times 9}, \tag{41}$$

$$\mathbf{M}_i = \begin{bmatrix} \mathbf{c}_i^m & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{c}_i^m & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{c}_i^m \end{bmatrix} \in \mathbb{R}^{3\times 9}. \tag{42}$$

The control term $\mathbf{u}_G$ is designed as in (31), based on a cost function $\gamma_G = \frac{1}{2}\|\mathbf{K}_3\mathbf{p} - \mathbf{C}_G\mathbf{h}_G\|^2 = \frac{1}{2}\mathbf{p}^\intercal\mathbf{A}_G\mathbf{p}$ by choosing $\mathbf{h}_G$ as an optimal least-squares parameter vector and with $\mathbf{A}_G = \mathbf{K}_3 - \mathbf{C}_H\mathbf{C}_H^+$. The scale and translation controllers are as in (33) and (34), respectively, with $s = h_s$ and $\mathbf{g} \in \mathbb{R}^3$. For controlling the rotations, note that $\mathbf{H} = h_s\mathbf{R}$ where

$$\mathbf{R} = \begin{bmatrix} 1 & -h_{rz}/h_s & h_{ry}/h_s \\ h_{rz}/h_s & 1 & -h_{rx}/h_s \\ -h_{ry}/h_s & h_{rx}/h_s & 1 \end{bmatrix}. \tag{43}$$

This can also be expressed as $\mathbf{R} = \mathbf{I}_3 + \mathbf{S}_r$, where $\mathbf{S}_r = [\frac{1}{h_s}\mathbf{h}_r]_\times$ is an antisymmetric matrix. Note that this $\mathbf{R}$ has the form of an infinitesimal rotation matrix and can approximate small-angle rotations. To define the desired formation orientation, we take $\mathbf{R}_d = \mathbf{I}_3$, without loss of generality[5]. We define the rotation control similarly to (35):

$$\mathbf{u}_R = -k_R\left(\mathbf{I}_N \otimes \mathbf{S}_r\right)\mathbf{C}_H\mathbf{h}_H. \tag{44}$$

The full control law is

$$\mathbf{u} = \mathbf{u}_H + \mathbf{u}_G + \mathbf{u}_s + \mathbf{u}_g + \mathbf{u}_R. \tag{45}$$

The discussion regarding implementation provided in Section IV-D is also valid for this 3D controller. Moreover, interestingly, we can provide a formal result similar to Theorem 1 of the 2D case: the following theorem, proven in the Appendix.

[5]For any $\mathbf{R}_d \in SO(3)$, one can redefine $\mathbf{c}_i = \mathbf{R}_d\mathbf{c}_i$ $\forall i \in \mathcal{N}$, and then redefine $\mathbf{R}_d = \mathbf{I}_3$.

Theorem 2. *If $h_s(t=0) > 0$, under the action of the control $\mathbf{u}_H + \mathbf{u}_G$ when* (17)*,* (31) *are computed in 3D space, $\mathbf{p}$ converges globally and exponentially fast to a static state for which the shape preservation condition is satisfied; i.e., $\forall i \in \mathcal{N}$, $\mathbf{p}_i = \mathbf{H}\mathbf{c}_i + \mathbf{t}$ for some $\mathbf{t} \in \mathbb{R}^3$ and an $\mathbf{H} \in \mathbb{R}^{3\times 3}$ having the form* (36) *with $h_s > 0$. In addition, $\mathbf{g}$ and $\mathbf{h}_H$ are invariant.*

The assumption $h_s(t=0) > 0$ in Theorem 2 is satisfied except for very largely rotated/deformed initial configurations. Therefore, it is reasonable to make this assumption in practical cases. We observe, similarly to the 2D case, invariance of γ_G under the scale, translation and rotation terms ($\mathbf{u}_s$, $\mathbf{u}_g$, and $\mathbf{u}_R$). Since now $\mathbf{H}$ is not a true rotation and scaling transform but, rather, an approximation for small rotations, some coupling between the controlled variables can appear. This effect does not prevent the suitable convergence of the controller and becomes smaller for smaller rotations.

B. Control with Kabsch algorithm

This uses the Kabsch algorithm as described in [27] for computing least-squares rotations, and hence does not rely on approximated expressions of rotations. On the other hand, since the Kabsch algorithm employs singular value decomposition, the formal analysis of the resulting controller is more challenging. We first compute the covariance matrix

$$\mathbf{F} = \mathbf{C}_b\mathbf{P}_b^\intercal \in \mathbb{R}^{3\times 3}, \tag{46}$$

where $\mathbf{C}_b = [\mathbf{c}_1, \ldots, \mathbf{c}_N]\,\mathbf{K} \in \mathbb{R}^{3\times N}$ and $\mathbf{P}_b = [\mathbf{p}_1, \ldots, \mathbf{p}_N]\,\mathbf{K} \in \mathbb{R}^{3\times N}$. Applying the singular value decomposition $\mathbf{F} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^\intercal$, the rotation matrix is

$$\mathbf{R} = \mathbf{V}\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & det(\mathbf{V}\mathbf{U}^\intercal) \end{bmatrix}\mathbf{U}^\intercal \in SO(3). \tag{47}$$

Following [12], we define the transformation matrix $\mathbf{H} = s\mathbf{R}$, where $s = tr(\mathbf{P}_b^\intercal\mathbf{R}\mathbf{C}_b)/tr(\mathbf{C}_b\mathbf{C}_b^\intercal)$ represents the scale. We can define a cost function for shape preservation as

$$\gamma_H = \frac{1}{2}\|\mathbf{P}_b - \mathbf{H}\mathbf{C}_b\|_F^2. \tag{48}$$

The corresponding gradient-based control term is as follows:

$$\mathbf{u}_H = -k_H \cdot vec\left(\mathbf{P}_b - \mathbf{H}\mathbf{C}_b\right). \tag{49}$$

The cost function and the term relative to the control using deformation modes are the same as with infinitesimal rotations. The translation control is formulated as in (34), with centroid $\mathbf{g} \in \mathbb{R}^3$, and the scale control is defined as

$$\mathbf{u}_s = k_s \cdot vec\left((s_d - s)\,(1/s)\,\mathbf{H}\mathbf{C}_b\right). \tag{50}$$

Finally, we use one of the control terms proposed in [12], which has the advantage of comprising scale and rotation in a simple manner and can achieve rotation control when simultaneously using (50) for scale control:

$$\mathbf{u}_{Hd} = -k_{Hd} \cdot vec\left(\mathbf{P}_b - \mathbf{H}_d\mathbf{C}_b\right), \tag{51}$$

where $\mathbf{H}_d = s_d\mathbf{R}_d$. The full controller is

$$\mathbf{u} = \mathbf{u}_H + \mathbf{u}_G + \mathbf{u}_s + \mathbf{u}_g + \mathbf{u}_{Hd}. \tag{52}$$

VI. Application of the control approach

Next, we discuss suitable task conditions for the application of our approach. Concrete examples are presented in Section VII.

- With the proposed controllers, the team can complete the transport task when the direct path from the current to the target configuration is traversable. If reaching the target configuration requires significant changes of direction along the way, the controller can also be used, by making the formation sequentially reach multiple predefined intermediate targets (i.e., waypoints).
- We can address tasks involving medium and large deformations. Such deformations may appear if (i) the initial configuration is highly deformed, or (ii) exogenous control actions (e.g., a reactive collision avoidance action at each robot) force the team to deform, due to encountering obstacles or a narrow passage. While other approaches may fail in such scenarios, our approach can succeed by allowing high adaptability of the shape while still maintaining a controlled deformation that keeps the integrity of the transported object.
- The proposed approach can also solve transport tasks requiring negligible or small deformations, i.e., rigid or mostly rigid transport. In these cases, our formulation allows for tight maintenance of the shape and scale while executing translations and rotations.
- The approach can be used to transport various types of objects. These include, for example, highly deformable planar objects (elastic sheets, fabrics), and fairly rigid and thicker objects such as mattresses.

VII. Experimental validation

We present several validation examples in 2D and 3D scenarios, including numerical simulations and experiments with physical mobile robots. Videos of these examples are also available through the provided link[6].

A. Experiments in 2D space

We report on tests done in two different platforms: Robotarium and CoppeliaSim. The modeled scenario in our tests is a typical practical one where the robots' grasping points on the object are in an elevated plane parallel to the ground [12], [13], [16]. We define error variables for position, scale and orientation as $\mathbf{e}_g = \mathbf{g} - \mathbf{g}_d$, $e_s = s - s_d$, $e_\theta = \theta - \theta_d$, respectively. Positions are expressed in meters and angles in radians. Therefore, γ_H and γ_G are expressed in squared meters, $\|\mathbf{e}_g\|$ in meters, and e_θ in radians. e_s is dimensionless.

1) Robotarium: First, we validate our 2D controller (1) using the Robotarium [30]. In this system, the control loop operates at 30 Hz both in simulation and in real-world execution. The robots in the real setup are localized with submillimeter precision by means of a motion capture system. As the robots are unicycle-type, we use the Robotarium's routines for converting the single-integrator velocities of our controller to unicycle velocities. Concretely, the conversion relies on a near-identity diffeomorphism. The robots have maximum linear and angular velocities of 0.2 m/s and 3.6 rad/s, respectively. To prevent drifts and oscillations due to non-ideal conditions, we make the robots stop when they get closer than 2 cm to their target positions. We also use the Robotarium's routines for collision avoidance (between agents, and with obstacles), which are based on barrier functions. To model the deformable objects, we implement the As-Rigid-As-Possible (ARAP) technique [31].

a) Simulations: We first conduct simulations for a formation of twelve robots manipulating a deformable sheet. The results are illustrated in Fig. 4. We consider a situation where a bent initial configuration has to be driven to an unbent target one. We test two cases. For Case 1 we choose $k_G = 0$. The team of robots quickly approaches the same shape as the reference configuration, but does so deforming the object in an uncontrolled way. For Case 2 we select $k_G = 10$. This allows the formation to maintain a linear and quadratic deformation. The movements are more efficient than in Case 1, producing gradual changes and staying close to a quadratic deformation pattern during the transient period. In both cases all errors converge to zero appropriately.

b) Robotic experiments: Now, we test the proposed controller in a real scenario with obstacles, where a team of eight robots transports a simulated sheet along a curved corridor to a target configuration (Fig. 5). To achieve the experiment's goal we manually define three intermediate waypoints, as a combination of shape, scale, position and orientation, to guide the team during the task. The first strategy we test is to control the formation scale, position and orientation, but not the shape. Therefore, all errors converge toward zero except those related to shape. The second strategy consists in applying $k_H = 0.2$ while maintaining $k_G = 0$. The formation then tends to preserve the shape of the object. However, as the term $\mathbf{u}_G$ is zero, the deformations, which appear inevitably due to the constraints of this particular scenario, are uncontrolled. In the third strategy, we use $k_H = 0.1$ and $k_G = 1$. In this case, the object deforms in a controlled way, preserving its integrity. It can be seen in the bottom-left image of Fig. 5 that the team's configuration undergoes a bending deformation pattern that allows it to better negotiate the turn in the corridor. This pattern corresponds to a quadratic deformation (see Fig. 2) and is not achievable when only linear deformation modes are used, as in [12], [13]. These results support the interest of using the linear and quadratic deformation modes to allow the object to undergo controlled deformations. Moreover, they demonstrate the applicability of the proposed approach on physical mobile robots.

2) CoppeliaSim: We now report on tests done in the realistic simulator CoppeliaSim, with the physics being modeled by the MuJoCo engine. We use six Pioneer 3-DX robots transporting a piece of fabric. Similarly to the tests above, the differential-drive kinematics of the robots are unicycle-type, and we map our controller's velocities accordingly. The defined transport task involves a translation of 3 m, a rotation of $\pi/2$ rad, and a scaling that reduces the distance between neighbor robots along the object's contour from 1.2 m to 1.1 m. Meanwhile, the team must also preserve its shape. We use

[6] https://tinyurl.com/deformvideolist

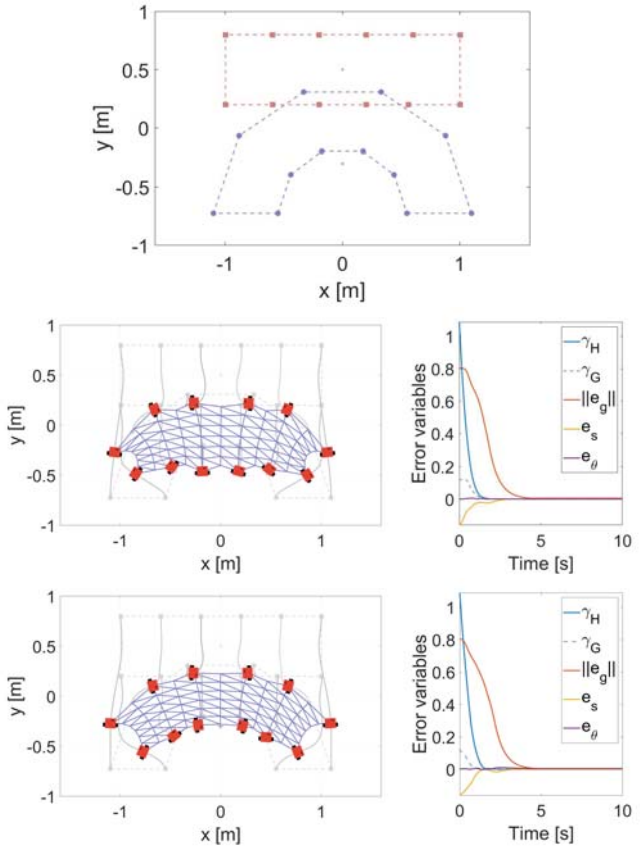


Fig. 4. Results in the Robotarium simulator. Top row: initial configuration (blue circles, bottom) and target configuration (red squares, top). Middle row: robot paths and error variables in Case 1. Bottom row: robot paths and error variables in Case 2. The configuration of the formation at instant $t = 0.8$ s is overlapped on the paths for both cases, showing that only in Case 2 the deformation tends to preserve a quadratic, bending-like pattern. In both cases, the gains are $k_H = 5$, $k_g = 0.5$, $k_s = 2$, and $k_\theta = 0.15$.

the gains $k_H = 2$, $k_G = 1$, $k_s = 1$, $k_g = 0.2$, $k_\theta = 0.2$. For comparison, we test on the same task a standard displacement-based formation control law that relies on consensus techniques [8, eq. 14]. We implement this alternative approach with an all-to-all formation graph (i.e., the same conditions as in our approach), and we add to it the centroid control term of our approach. The cycle time of the control law loop is 0.01 ms. To prevent drift and oscillation near the target positions, we stop applying the controller when the error goes below a threshold of 2 cm from these positions. The results are illustrated in Fig. 6. The proposed approach maintains all variables in proper value ranges, which clearly leads to the integrity of the object being preserved. The alternative approach also completes the task and keeps most variables in suitable ranges. However, it does not provide close control of all of them; one can observe from the intermediary configuration and the evolution of e_s that the scale reaches excessively low values during the transport, which could compromise the safety of the system. In particular, the lowest scale value reached is 29% below the desired one, while with our approach this is only 0.2%.

B. Experiments in 3D space

We validate our 3D controllers in numerical MATLAB simulations. The simulation time step we use is 0.01 s. The error for the orientation is now defined by the dimensionless variable $e_R = \|\mathbf{I}_3 - \mathbf{R}\mathbf{R}_d^{-1}\|_F$. We first show an example where a team of eight robots transports an ARAP-modelled deformable object from an initial deformed configuration to a rectangular target configuration with a different orientation in 3D space. We choose an example where a considerable rotation is needed, to test the capabilities of the two proposed approaches for rotation representation. In addition, the chosen initial configuration has a large deformation. We use saturation of the Euclidean norm of the robots' velocities at 1.5 m/s, to model realistic motion constraints. The results are illustrated in Fig. 7. All errors converge to zero and the behavior is satisfactory for both controllers; this was also the case in tests of the 3D controller that we conducted in other scenarios. In part (c) of the figure, where we are handling scale and orientation jointly (see (51)), γ_H exhibits a mild oscillation in the first seconds. In part (b), the degree of uncoupling between the error variables is higher.

We repeat the experiment with infinitesimal rotations, with white Gaussian noise of different standard deviation ($\sigma = 1$ mm and $\sigma = 10$ mm) being added to p_{ix}, p_{iy} and p_{iz} for every robot i. The results, illustrated in Fig. 8, show that the controller can withstand these perturbations. The magnitude of the position error after reaching a perturbed steady-state regime is proportional to the magnitude of the perturbation.

As noted in Section VI, these controllers can also be used in situations where complete preservation of some of the variables is required; e.g., for rigid transport. We demonstrate this with an example where a team of six robots transports a mattress along a curved corridor in 3D space. We model this object with the meshless shape matching (MSM) method in [22] and use the control with the Kabsch algorithm. The results are illustrated in Fig. 9. We choose k_H and k_s to be much higher than the rest of the gains. As a result, the shape and the scale are preserved during the transport, as is desired for this task.

As discussed in Section IV-D, our approach is simple computationally and can scale well for typical team sizes (N), even when implemented in a centralized manner. We verify this fact by measuring the computational cost of the three control laws (1), (45), and (52), running them in MATLAB on a computer with an 11th-generation Intel Core i7 CPU at 2.80 GHz and 16 GB RAM. In the worst case, the average computation time per control loop cycle remains below 0.02 ms for $N = 10$, 0.2 ms for $N = 50$, and 20 ms for $N = 500$.

C. Discussion

The presented experimental results showcase the versatility of our approach, which can successfully handle diverse tasks, workspaces (2D and 3D), robot motion models, and objects (various deformable bodies modeled with ARAP, MuJoCo, and MSM). The selection of control gains in the reported tests was guided by the desired behavior; dominant terms were assigned higher gains, and simple adjustments based

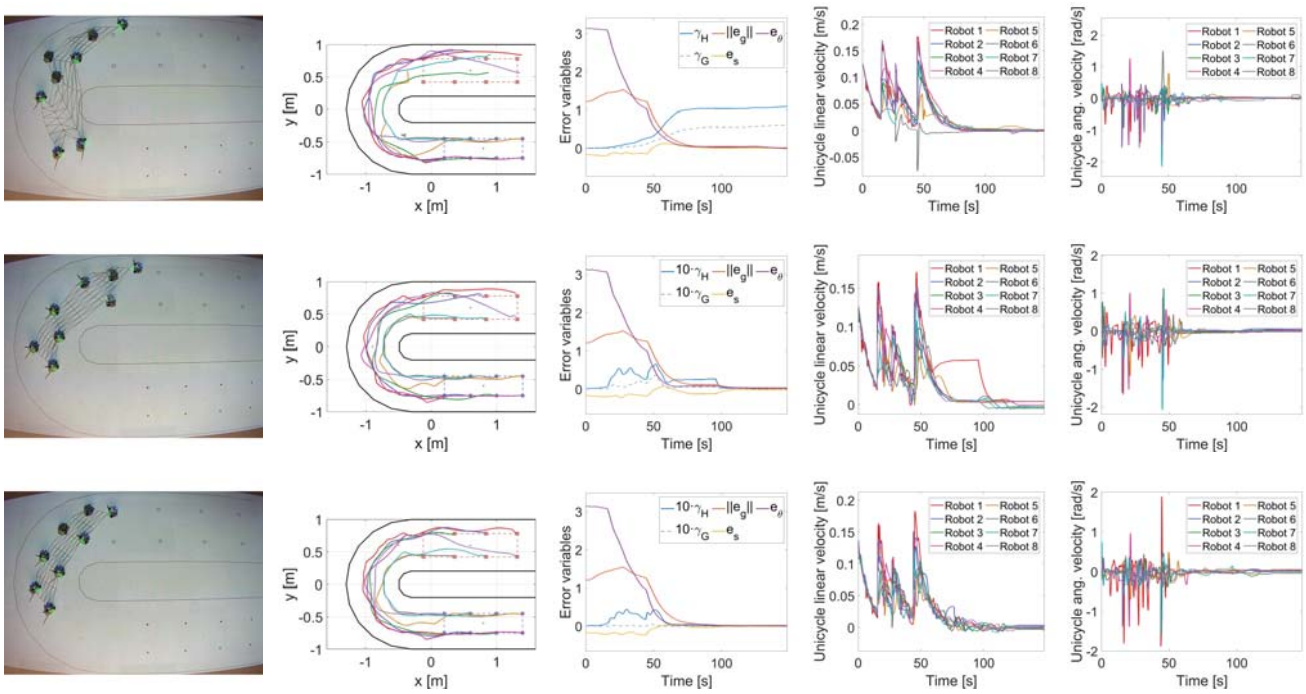


Fig. 5. Experimental results in the Robotarium. For each row, the five plots from left to right are: representative top-view snapshots, robot paths, control errors, unicycle linear velocities, and unicycle angular velocities. Each row corresponds to a different shape-control strategy: 1st, the shape is not controlled in any way, $k_H = k_G = 0$, and the shape errors (γ_H, γ_G) do not converge toward zero; 2nd, deformation is not controlled, $k_H \neq 0$ and $k_G = 0$; and 3rd, deformation is controlled, $k_H \neq 0$ and $k_G \neq 0$. In the 2nd and 3rd cases, although the shape cannot be preserved, all the errors reach zero values, but only in the 3rd strategy the object deforms in a controlled manner. The improvement of the variables in the 3rd case with respect to the 2nd one can be seen in the error evolution between 50–100 s. The values of the remaining gains are $k_g = k_s = k_\theta = 0.1$ in the three tests.

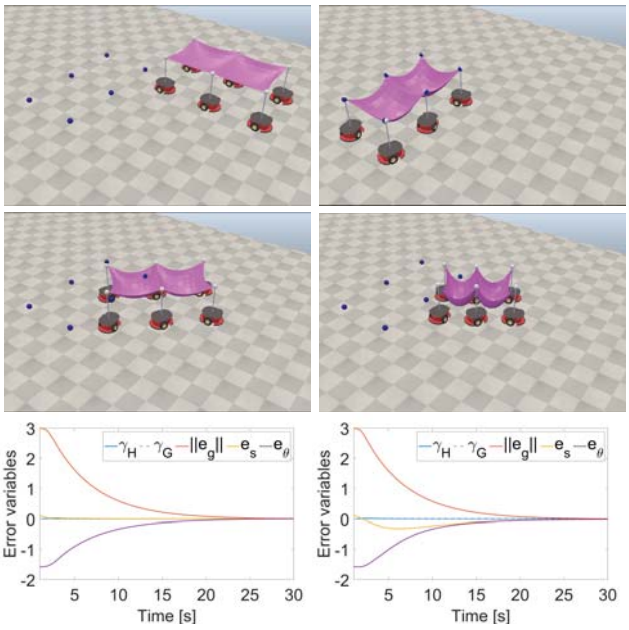
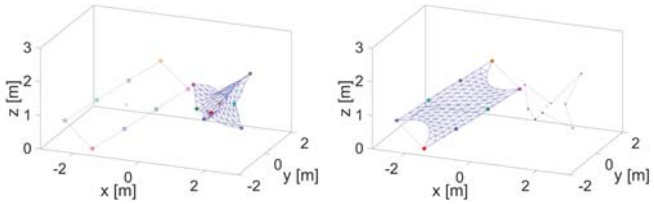


Fig. 6. Simulation results in CoppeliaSim. Top row: initial (left) and final (right) configuration of the task. The positions of the grasping points corresponding to the target configuration are shown as blue spheres. Second row: two intermediary configurations when using the proposed approach (left) and the alternative approach (right). Bottom row: error variables with the proposed approach (left) and with the alternative approach (right).

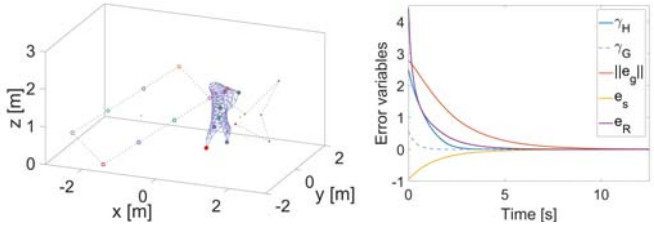
on observed execution results were made if required. Our approach keeps the integrity of the object by closely controlling the team's configuration, as mentioned in Section III and demonstrated by the presented experimental results. However, we do not provide explicit quantitative guarantees in this respect. As mentioned in the introduction, transporting objects with non-uniform material properties would require a different approach; the standard geometric deformation modes we use may not adapt well to such objects. Collisions and other safety hazards are not explicitly considered in our approach. Another issue we do not address is the automatic definition of intermediate waypoints during the transport; this type of planning problem exceeds the scope of our article.

VIII. Conclusion

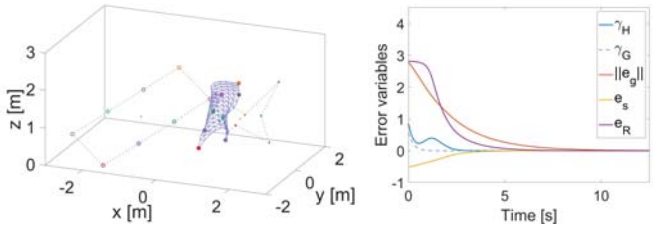
We have presented a novel multirobot formation control approach to transport deformable objects that allows high flexibility while maintaining a controlled deformation. This is achieved by exploiting deformation modes to constrain the patterns of deformation that the object can undergo during the transport. The approach is supported by formal analysis and its performance has been validated by simulations and experiments in diverse 2D and 3D scenarios. In our next steps, we wish to consider extensions of this approach such as: more complex robot dynamics, a fully developed distributed implementation, the explicit handling of safety requirements, and the use of time-varying target configurations.



(a) Initial (left) and target (right) configurations of the formation

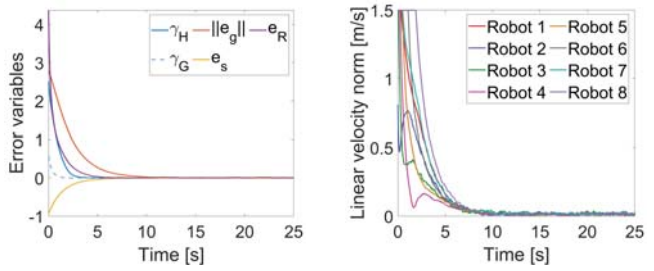


(b) Control with infinitesimal rotations

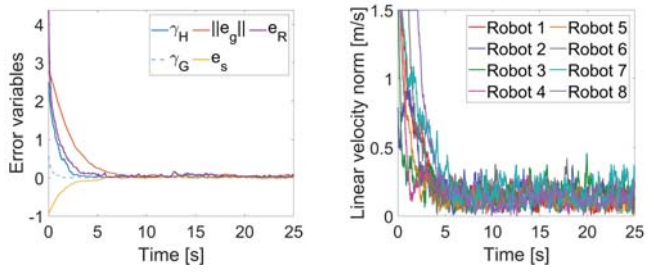


(c) Control with Kabsch algorithm

Fig. 7. Simulation results in 3D space. For (b) and (c), the plots are: (Left) Configuration of the formation at instant $t = 0.8$ s; (Right) Evolution of the error variables. In both cases, the gains are $k_H = 1$, $k_G = 2$, $k_g = 0.5$, $k_s = 0.5$, and $k_R = k_{Hd} = 0.8$.



(a) Noise σ = 1 mm



(b) Noise σ = 10 mm

Fig. 8. Simulation results in 3D space with added noise. (Left) Evolution of the error variables; (Right) Evolution of the Euclidean norms of the robots' velocities, i.e., $\|\mathbf{u}_i\|$ $\forall i \in \mathcal{N}$.

DISCLOSURE STATEMENT

The authors report no potential conflict of interest.

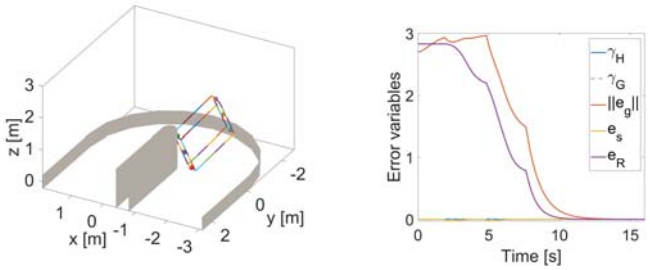


Fig. 9. (Left) Configuration of the formation at instant $t = 6$ s; (Right) Evolution of the error variables. The gains are $k_H = 50$, $k_G = 5$, $k_g = 1$, $k_s = 50$, and $k_{Hd} = 1.5$.

DATA AVAILABILITY STATEMENT

The data that support the findings of this study are available from the corresponding author, M. A., upon reasonable request.

ACKNOWLEDGEMENTS

This work was supported via project REMAIN - S1/1.1/E0111 (Interreg Sudoe Programme, ERDF), and via projects PID2021-124137OB-I00 and TED2021-130224B-I00 funded by MCIN/AEI/10.13039/501100011033, by ERDF A way of making Europe and by the European Union NextGenerationEU/PRTR. R. Marcos-Saavedra was supported by Cátedra de Transformación Industrial (Universidad de Zaragoza - Gobierno de Aragón). M. Aranda was supported via a María Zambrano Fellowship funded by the Spanish Ministry of Universities and by the European Union-NextGenerationEU.

APPENDIX

A. Linear least-squares problem

In the problem $\arg\min_{\mathbf{x}\in\mathbb{R}^n} \|\mathbf{A}\mathbf{x}-\mathbf{b}\|$ for given $\mathbf{A} \in \mathbb{R}^{m\times n}$ and $\mathbf{b} \in \mathbb{R}^m$, $\mathbf{x} = \mathbf{A}^+\mathbf{b}$ is the solution for which $\|\mathbf{x}\|$ is minimum, and it is the only solution if the rank of $\mathbf{A}$ is n [32]. In all cases, $\mathbf{b}_A = \mathbf{A}\mathbf{A}^+\mathbf{b}$ is the unique orthogonal projection of $\mathbf{b}$ onto the column space, or range, of $\mathbf{A}$ [28, ch. III]. In all the instances of this problem considered in this article, the team configuration resulting from solving the problem corresponds to this orthogonal projection.

B. Proof of Theorem 1

To study the dynamic evolution of the team's shape, we define $\mathbf{A}_{HG} = k_H\mathbf{A}_H + k_G\mathbf{A}_G$. We have

$$\dot{\mathbf{p}} = \mathbf{u}_H + \mathbf{u}_G = -\mathbf{A}_{HG}\,\mathbf{p}. \tag{53}$$

As both $\mathbf{A}_H$ and $\mathbf{A}_G$ are symmetric and positive semidefinite and the gains k_H and k_G are positive, $\mathbf{A}_{HG}$ is symmetric and positive semidefinite. Hence, from standard results in the theory of linear dynamical systems, we know that $\|\mathbf{p}\|$ remains bounded and $\mathbf{p}$ converges asymptotically, exponentially fast, to a static state that corresponds to the orthogonal projection of $\mathbf{p}(t = 0)$ onto the kernel, or nullspace, of $\mathbf{A}_{HG}$. We have $\mathbf{A}_{HG}\mathbf{p}(t \to \infty) = \mathbf{0}$ and hence $\mathbf{p}(t \to \infty)^\intercal\mathbf{A}_{HG}\mathbf{p}(t \to \infty) = 0$ which, since $\gamma_H(\mathbf{p}) \geq 0$ and $\gamma_G(\mathbf{p}) \geq 0$ for all $\mathbf{p} \in \mathbb{R}^{2N}$, implies $\gamma_H(t \to \infty) = 0$ and $\gamma_G(t \to \infty) = 0$.

Next, we study the evolution of $\mathbf{g}$. Recall that the centroids of $(\mathbf{p}_i)_{i=1}^N$, $(\mathbf{p}_{H,i})_{i=1}^N$, and $(\mathbf{p}_{G,i})_{i=1}^N$ are equal. Therefore, from (18) and (32) one can see that the sums, over all i, of the x and y velocity components due to $\mathbf{u}_H + \mathbf{u}_G$ are always zero. This implies the centroid $\mathbf{g}$ is invariant. We now study $\dot{\mathbf{h}}_H$. Recall that $\mathbf{C}_H^+ = (\mathbf{C}_H^\intercal \mathbf{C}_H)^+ \mathbf{C}_H^\intercal$; using that $\mathbf{C}_H = \mathbf{K}_2 \mathbf{C}_H$ and $\mathbf{C}_H^\intercal = \mathbf{C}_H^\intercal \mathbf{K}_2$ and simple operations, we can write

$$\dot{\mathbf{h}}_H = \mathbf{C}_H^+ \dot{\mathbf{p}} = -(\mathbf{C}_H^\intercal \mathbf{C}_H)^+ \Big(k_H \mathbf{C}_H^\intercal (\mathbf{I}_{2N} - \mathbf{C}_H \mathbf{C}_H^+) + k_G \mathbf{C}_H^\intercal (\mathbf{I}_{2N} - \mathbf{C}_G \mathbf{C}_G^+) \Big) \mathbf{p}. \quad (54)$$

The two columns of $\mathbf{C}_H$ are in the column space of $\mathbf{C}_G$ because $col_1(\mathbf{C}_H) = col_1(\mathbf{C}_G) + col_4(\mathbf{C}_G)$ and $col_2(\mathbf{C}_H) = -col_2(\mathbf{C}_G) + col_3(\mathbf{C}_G)$. Therefore, as $\mathbf{C}_G \mathbf{C}_G^+$ projects onto the column space of $\mathbf{C}_G$, we have $(\mathbf{C}_G \mathbf{C}_G^+)\mathbf{C}_H = \mathbf{C}_H$. The same fact applies trivially to $\mathbf{C}_H \mathbf{C}_H^+$ too. Hence, we have

$$(\mathbf{I}_{2N} - \mathbf{C}_H \mathbf{C}_H^+)\mathbf{C}_H = \mathbf{0},\ (\mathbf{I}_{2N} - \mathbf{C}_G \mathbf{C}_G^+)\mathbf{C}_H = \mathbf{0}, \quad (55)$$

which, due to the symmetry of the matrices, yields

$$\mathbf{C}_H^\intercal (\mathbf{I}_{2N} - \mathbf{C}_H \mathbf{C}_H^+) = \mathbf{0},\ \mathbf{C}_H^\intercal (\mathbf{I}_{2N} - \mathbf{C}_G \mathbf{C}_G^+) = \mathbf{0}. \quad (56)$$

Substituting into (54) gives $\dot{\mathbf{h}}_H = \mathbf{0}$; i.e., $\mathbf{h}_H$ is invariant. This implies that $s(t) > 0$ and hence $\mathbf{H}(t) \neq \mathbf{0}$ for all $t \geq 0$.

The fact that $\gamma_H(t \to \infty) = 0$ implies, from (16), that as $t \to \infty$, $\mathbf{K}_2 \mathbf{p} \to \mathbf{C}_H \mathbf{h}_H$. This expression can be broken down for every i as in (5), giving $\mathbf{p}_i \to \mathbf{p}_{H,i}\ \forall i \in \mathcal{N}$. Moreover, we have that $\mathbf{H}(t \to \infty) \neq \mathbf{0}$. These facts imply that the defined shape preservation condition is satisfied; i.e., $\mathbf{p}_i(t \to \infty) = \mathbf{H}\mathbf{c}_i + \mathbf{t}\ \forall i \in \mathcal{N}$, for some $\mathbf{t} \in \mathbb{R}^2$ and a non-zero $\mathbf{H} \in \mathbb{R}^{2\times 2}$ having the form in (2). ■

C. Proof of Proposition 1

Notice that one can write

$$\dot{\gamma}_H = \mathbf{p}^\intercal \mathbf{A}_H \dot{\mathbf{p}},\ \ \dot{\gamma}_G = \mathbf{p}^\intercal \mathbf{A}_G \dot{\mathbf{p}}. \quad (57)$$

Therefore, we study the invariance by replacing $\dot{\mathbf{p}}$ in these equations by each of the three considered control terms: $\mathbf{u}_s$, $\mathbf{u}_g$, and $\mathbf{u}_\theta$. In particular, we will show that $\mathbf{A}_H \dot{\mathbf{p}} = \mathbf{0}$ and $\mathbf{A}_G \dot{\mathbf{p}} = \mathbf{0}$ for each of the three terms, which, according to (57), implies the invariance of γ_H and γ_G. In particular, consider the following six equations:

$$\mathbf{A}_H \mathbf{C}_H = \mathbf{0},\ \ \mathbf{A}_G \mathbf{C}_H = \mathbf{0}, \quad (58)$$

$$\mathbf{A}_H (\mathbf{I}_N \otimes \mathbf{S}) \mathbf{C}_H = \mathbf{0},\ \ \mathbf{A}_G (\mathbf{I}_N \otimes \mathbf{S}) \mathbf{C}_H = \mathbf{0}, \quad (59)$$

$$\mathbf{A}_H (\mathbf{1}_N \otimes (\mathbf{g}_d - \mathbf{g})) = \mathbf{0},\ \ \mathbf{A}_G (\mathbf{1}_N \otimes (\mathbf{g}_d - \mathbf{g})) = \mathbf{0}. \quad (60)$$

One can directly see from the expressions of the control terms $\mathbf{u}_s$ in (33), $\mathbf{u}_g$ in (34), and $\mathbf{u}_\theta$ in (35), that the statement we want to prove is true if the six equations in (58) to (60) are satisfied. Therefore, we now proceed to show that these equations are satisfied. First, notice that left-multiplying by $\mathbf{K}_2$ in the two equations of (55), we directly obtain (58). Then, denote $\mathbf{C}_H^\perp = (\mathbf{I}_N \otimes \mathbf{S})\mathbf{C}_H$ and observe that $\mathbf{C}_H^\perp = [col_2(\mathbf{C}_H), -col_1(\mathbf{C}_H)]$, which means $\mathbf{C}_H^\perp$ has the same columns as $\mathbf{C}_H$ up to order and sign. Therefore, using the arguments in the proof of Theorem 1, the two columns of $\mathbf{C}_H^\perp$ are in the column space of both $\mathbf{C}_H$ and $\mathbf{C}_G$, and we have

$$(\mathbf{I}_{2N} - \mathbf{C}_H \mathbf{C}_H^+)\mathbf{C}_H^\perp = \mathbf{0},\ (\mathbf{I}_{2N} - \mathbf{C}_G \mathbf{C}_G^+)\mathbf{C}_H^\perp = \mathbf{0}. \quad (61)$$

Again, left-multiplying by $\mathbf{K}_2$, we obtain (59). To conclude, we will address the equations in (60). Since $\mathbf{K}_2 \mathbf{A}_H = \mathbf{A}_H$ and this matrix is symmetric, the expression $\mathbf{A}_H = \mathbf{A}_H \mathbf{K}_2$ holds true. Using this expression, one can write the left-hand side of the first equation in (60) as $\mathbf{A}_H$ left-multiplying the expression $\mathbf{K}_2(\mathbf{1}_N \otimes (\mathbf{g}_d - \mathbf{g}))$. In this expression, the product by $\mathbf{K}_2$ subtracts the mean from a $2N \times 1$ vector of N positions in 2D which are all equal: therefore, the result is the zero vector. The same reasoning applies with $\mathbf{A}_G$ in place of $\mathbf{A}_H$. Consequently, the equations in (60) are satisfied, which means the proof is concluded. ■

D. Proof of Theorem 2

One can follow similar steps to those employed in the proof of Theorem 1. Notice that in the 3D case, it still holds that $\mathbf{A}_H$ and $\mathbf{A}_G$ are symmetric and positive semidefinite. One can define $\mathbf{A}_{HG}$ in the same way as before and conclude that $\mathbf{p}$ converges asymptotically, exponentially fast, to a static state corresponding to the orthogonal projection of $\mathbf{p}(t=0)$ onto the kernel of $\mathbf{A}_{HG}$. This implies that $\gamma_H(t \to \infty) = 0$. The invariance of the centroid $\mathbf{g} \in \mathbb{R}^3$ can also be proven with the same arguments as in Theorem 1. Moreover, to study the invariance of $\mathbf{h}_H \in \mathbb{R}^4$, one can consider the equation (54) as in the 2D case, and notice that in the 3D case it holds that

$$col_1(\mathbf{C}_H) = col_1(\mathbf{C}_G) + col_5(\mathbf{C}_G) + col_9(\mathbf{C}_G), \quad (62)$$

$$col_2(\mathbf{C}_H) = -col_6(\mathbf{C}_G) + col_8(\mathbf{C}_G), \quad (63)$$

$$col_3(\mathbf{C}_H) = col_3(\mathbf{C}_G) - col_7(\mathbf{C}_G), \quad (64)$$

$$col_4(\mathbf{C}_H) = -col_2(\mathbf{C}_G) + col_4(\mathbf{C}_G). \quad (65)$$

This implies that every column of $\mathbf{C}_H$ is in the column space of $\mathbf{C}_G$. Therefore, using the same deductive process as in the 2D case, it can be concluded that $\mathbf{h}_H$ is invariant. Since $\gamma_H(t \to \infty) = 0$, $h_s(t=0) > 0$, and $\mathbf{h}_H$ is invariant, the robot positions $\mathbf{p}_i$ converge to a configuration satisfying $\mathbf{K}_3 \mathbf{p} = \mathbf{C}_H \mathbf{h}_H$ with $h_s > 0$. This again implies that the shape preservation condition for the 3D case is satisfied; i.e., $\mathbf{p}_i(t \to \infty) = \mathbf{H}\mathbf{c}_i + \mathbf{t}\ \forall i \in \mathcal{N}$, for some $\mathbf{t} \in \mathbb{R}^3$ and an $\mathbf{H} \in \mathbb{R}^{3\times 3}$ having the form in (36) with $h_s > 0$. ■

REFERENCES

[1] H. Farivarnejad and S. Berman, "Multirobot control strategies for collective transport," *Annual Review of Control, Robotics, and Autonomous Systems*, vol. 5, pp. 205-219, 2022.

[2] Z. Wang and M. Schwager, "Force-amplifying n-robot transport system (Force-ANTS) for cooperative planar manipulation without communication," *Int. J. Robot. Res.*, vol. 35, no. 13, pp. 1564-1586, 2016.

[3] D. Koung, O. Kermorgant, I. Fantoni and L. Belouaer, "Cooperative multi-robot object transportation system based on hierarchical quadratic programming," *IEEE Robot. Automat. Lett.*, vol. 6, no. 4, pp. 6466-6472, 2021.

[4] H. Yin, A. Varava and D. Kragic, "Modeling, learning, perception, and control methods for deformable object manipulation," *Sci Robot.* vol. 6, no. 54, art. eabd8803, 2021.

[5] R. Herguedas, G. López-Nicolás, R. Aragüés and C. Sagüés, "Survey on multi-robot manipulation of deformable objects," in *Proc. IEEE Int. Conf. Emerg. Technol. Fact. Autom.*, 2019, pp. 977-984.

[6] Z. Peng, G. Wen, A. Rahmani and Y. Yu, "Distributed consensus-based formation control for multiple nonholonomic mobile robots with a specified reference trajectory," *Int. J. Syst. Sci.*, vol. 46, no. 8, pp. 1447–1457, 2013.
[7] Z. Lin, L. Wang, Z. Han and M. Fu, "Distributed formation control of multi-agent systems using complex Laplacian," *IEEE Trans. Autom. Control*, vol. 59, no. 7, pp. 1765-1777, 2014.
[8] K.-K. Oh, M.-C. Park and H.-S. Ahn, "A survey of multi-agent formation control," *Automatica*, vol. 53, pp. 424-440, 2015.
[9] M. Aranda, R. Aragüés and G. López-Nicolás, "Combined leaderless control of translational, shape-preserving, and affine multirobot formations," in *IEEE Robot. Automat. Lett.*, vol. 8, no. 11, pp. 7567-7574, 2023.
[10] H. Bai and J. T. Wen, "Cooperative load transport: A formation-control perspective," *IEEE Trans. Robot.*, vol. 26, no. 4, pp. 742-750, 2010.
[11] J. Alonso-Mora, R. Knepper, R. Siegwart and D. Rus, "Local motion planning for collaborative multi-robot manipulation of deformable objects," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2015, pp. 5495-5502.
[12] M. Aranda, J. Sanchez, J. A. Corrales Ramon and Y. Mezouar, "Robotic motion coordination based on a geometric deformation measure," *IEEE Syst. J.*, vol. 16, no. 3, pp. 3689-3699, 2022.
[13] R. Herguedas, M. Aranda, G. López-Nicolás, C. Sagüés and Y. Mezouar, "Double-integrator multirobot control with uncoupled dynamics for transport of deformable objects," *IEEE Robot. Automat. Lett.*, vol. 8, no. 11, pp. 7623-7630, 2023.
[14] D. Kruse, R. J. Radke and J. T. Wen, "Collaborative human-robot manipulation of highly deformable materials," *Proc. 2015 IEEE Int. Conf. Robot. Autom. (ICRA)*, pp. 3782-3787, 2015.
[15] B. Aksoy and J. T. Wen, 'Collaborative manipulation of deformable objects with predictive obstacle avoidance," in *Proc. 2024 IEEE Int. Conf. Robot. Autom. (ICRA)*, pp. 16766-16772, 2024.
[16] J. Hu, W. Liu, H. Zhang, J. Yi and Z. Xiong, "Multi-robot object transport motion planning with a deformable sheet," *IEEE Robot. Automat. Lett.*, vol. 7, no. 4, pp. 9350-9357, 2022.
[17] C. Yang, G. N. Sue, Z. Li, L. Yang, H. Shen, Y. Chi, A. Rai, J. Zeng and K. Sreenath, "Collaborative navigation and manipulation of a cable-towed load by multiple quadrupedal robots," *IEEE Robot. Automat. Lett.*, vol. 7, no. 4, pp. 10041-10048, 2022.
[18] R. Cotsakis, D. St-Onge and G. Beltrame, "Decentralized collaborative transport of fabrics using micro-UAVs," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2019, pp. 7734-7740.
[19] J. Gimenez, L. R. Salinas, D. C. Gandolfo, C. D. Rosales and R. Carelli, "Control for cooperative transport of a bar-shaped payload with rotorcraft UAVs including a landing stage on mobile robots," *Int. J. Syst. Sci.*, vol. 51, no. 16, pp. 3378-3392, 2020.
[20] G. Li, X. Liu and G. Loianno, "RotorTM: A flexible simulator for aerial transportation and manipulation," *IEEE Trans. Robot.*, vol. 40, pp. 831-850, 2024.
[21] A. Pentland and J. Williams, "Good vibrations: modal dynamics for graphics and animation," *SIGGRAPH Comput. Graph.*, vol. 23, no. 3, pp. 207-214, 1989.
[22] M. Müller, B. Heidelberger, M. Teschner and M. Gross, "Meshless deformations based on shape matching," in *ACM Trans. Graph.*, vol. 24, pp. 471-478, 2005.
[23] P. Güler, A. Pieropan, M. Ishikawa and D. Kragic, "Estimating deformability of objects using meshless shape matching," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, pp. 5941-5948, 2017.
[24] B. Yang, B. Lu, W. Chen, F. Zhong and Y.-H. Liu, "Model-free 3-D shape control of deformable objects using novel features based on modal analysis," *IEEE Trans. Robot.*, vol. 39, no. 4, pp. 3134-3153, 2023.
[25] R. Marcos-Saavedra, M. Aranda and G. López-Nicolás, "Multirobot transport of deformable objects using deformation modes," *4th Workshop on Repres. and Manipul. Deform. Obj., 2024 IEEE Int. Conf. Robot. Autom.*, Available at: https://deformable-workshop.github.io/icra2024/spotlight/02_03_wdo_marcossaavedra_multirobot.pdf, 2024.
[26] O. Sorkine, D. Cohen-Or, Y. Lipman, M. Alexa, C. Rössl and H.-P. Seidel, "Laplacian surface editing," in *Proc. 2004 Eurographics/ACM SIGGRAPH Symp. Geometry Processing*, 2004, pp. 175-184.
[27] O. Sorkine-Hornung and M. Rabinovich, "Least-squares rigid motion using SVD," *Tech. Rep. ETH Zurich*, Available at: https://igl.ethz.ch/projects/ARAP/svd_rot.pdf, 2017.
[28] A. Albert, "Regression and the Moore-Penrose pseudoinverse," *Book Series Math. Sci. Eng*, vol. 94, Elsevier, 1972.
[29] S. S. Kia, B. Van Scoy, J. Cortes, R. A. Freeman, K. M. Lynch and S. Martinez, "Tutorial on dynamic average consensus: the problem, its applications, and the algorithms," *IEEE Control Syst. Mag.*, vol. 39, no. 3, pp. 40-72, 2019.
[30] S. Wilson, P. Glotfelter, L. Wang, S. Mayya, G. Notomista, M. Mote and M. Egerstedt, "The Robotarium: Globally impactful opportunities, challenges, and lessons learned in remote-access, distributed control of multirobot systems," *IEEE Control Syst. Mag.*, vol. 40, no. 1, pp. 26-44, 2020.
[31] O. Sorkine and M. Alexa, "As-rigid-as-possible surface modeling," in *Proc. 2007 Eurographics Symp. Geometry Process.*, 2007, pp. 109-116.
[32] M. Planitz, "Inconsistent systems of linear equations," *Math. Gaz.*, vol. 63, no. 425, pp. 181-85, 1979.

PLACE PHOTO HERE

Raquel Marcos-Saavedra graduated in industrial technology engineering and then received the Masters degree in industrial engineering from Universidad de Zaragoza, Spain, in 2024. Her research interests are mainly centered on the control of multirobot systems.

PLACE PHOTO HERE

Miguel Aranda received the Ph.D. degree in systems engineering and computer science from Universidad de Zaragoza, Spain, in 2015. He held postdoctoral research positions with Institut Pascal Laboratory, Clermont-Ferrand, France. He is currently a Research Fellow with Instituto de Investigación en Ingeniería de Aragón (I3A), Universidad de Zaragoza. His current research interests include multiagent control and robotic manipulation.

PLACE PHOTO HERE

Gonzalo López-Nicolás received the Ph.D. degree in systems engineering and computer science from the University of Zaragoza, Spain, in 2008. He is currently a Full Professor with the Department of Computer Science and Systems Engineering, University of Zaragoza. He is also a member of the Robotics, Computer Vision and Artificial Intelligence Group, and the Aragon Institute of Engineering Research (I3A). His current research interests include visual control, autonomous robot navigation, multirobot systems, and the application of computer vision techniques to robotics and assistive devices.