Robotic motion coordination based on a geometric deformation measure

Miguel Aranda, Jose Sanchez, Juan Antonio Corrales Ramon and Youcef Mezouar

***Abstract*—This article describes a novel approach to achieve motion coordination in a multirobot system based on the concept of deformation. Our main novel contribution is to link these two elements (namely, coordination and deformation). In particular, the core idea of our approach is that the robots' motions minimize a global measure of the deformation of their positions relative to a prescribed shape. Based on this idea we propose a linear shape controller, that also incorporates a term modeling an affine deformation. We show that the affine term is particularly useful when the deformation to be controlled is large. We also propose controls for the other variables (centroid, rotation, size) that define the geometric configuration of the team. Importantly, these additional controls are completely decoupled from the shape control. The overall approach is simple and robust, and it creates closely coordinated robot motions. Being based on deformation, it is useful in several scenarios involving manipulation tasks: e.g., handling of a highly deformable object, control of an object's shape, or regulation of the shape formed by the fingertips of a robotic hand. We present simulation and experimental results to validate the proposed approach.**

***Index Terms*—Multirobot systems, motion coordination, cooperative manipulation, robot control.**

I. Introduction

A. Overall context

Recent technological advances in artificial intelligence (AI) are bringing about higher capabilities to solve problems of societal interest. Robotic systems are a particular manifestation of artificial intelligence in which such intelligence becomes embodied. Specifically, aside from being capable of processing information, a robot also has the ability to move and act autonomously in the world. This gives it great potential to carry out many tasks. For high-complexity tasks, a multirobot system provides more advanced capabilities (coverage of larger workspaces, handling of higher payloads, better resilience via redundancy) than a single robot [1].

A salient topic in present-day robotics is the handling of *deformation*. Robotics has traditionally concentrated on dealing with *rigid* objects. Deformable objects, on the other hand, are interesting in many applications, but it is very challenging to perceive, model and manipulate them with robots. The recent survey [2] details the associated challenges. Using a multirobot system (instead of a single robot) helps to mitigate some of these challenges, due to the higher number of resources and of degrees of freedom provided by such a system [3]. As a result, a problem of current interest is to develop appropriate strategies to coordinate the motions in a multirobot system to address tasks that involve deformation. This is the problem addressed in our work.

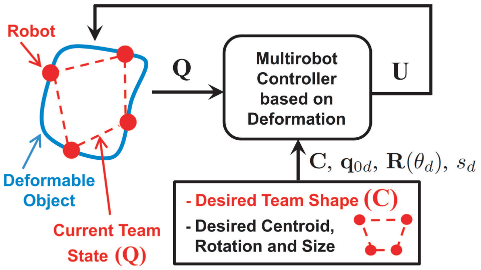
Robot
Q
Multirobot Controller based on Deformation
U
Deformable Object
Current Team State (Q)
$\mathbf{C}$, $\mathbf{q}_{0d}$, $\mathbf{R}(\theta_d)$, s_d
- Desired Team Shape (C)
- Desired Centroid, Rotation and Size

Fig. 1. Diagram representing the proposed approach with four robots.

B. Contribution and method description

Our main novel contribution is to use a geometric deformation measure directly as the cost function for controlling the motion of a multirobot system. As a result, we link coordination and deformation. To see why this link is interesting, consider a scenario where multiple robots manipulate a deformable object. Then: *(i) coordination without considering deformation* (e.g., multirobot formation control [4]) means the state of the team is controlled appropriately but, as deformation information is disregarded, the chances of damaging the object during the motion increase. On the other hand, *(ii) deformation without considering coordination* (e.g., object deformation control) means each robot controls individually the deformation, but the lack of explicit coordination between the robots' efforts implies their actions are suboptimal and more likely to fall into local minima [5].

A block diagram of our approach appears in Fig. 1. The deformation measure we use is defined as the difference in shape between the current and the desired set of robot positions. The core goal of our approach is to control the robots towards maintaining the desired shape. To this end, we propose a *shape controller*, consisting of two control terms: a pure deformation term, and an affine deformation term. The pure deformation term makes the robots minimize the deformation measure, by moving them towards the optimal

M. Aranda, J. Sanchez and Y. Mezouar are with CNRS, Clermont Auvergne INP, Institut Pascal, Université Clermont Auvergne, F-63000 Clermont-Ferrand, France. J. A. Corrales Ramon is with Centro Singular de Investigación en Tecnoloxías Intelixentes (CiTIUS), Universidade de Santiago de Compostela, 15782 Santiago de Compostela, Spain. Corresponding author: Miguel Aranda, email: miguel.aranda@sigma-clermont.fr.

This work was supported by project SOFTMANBOT, which received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 869855; project CoMManDIA (SOE2/P1/F0638), which is cofinanced by Interreg Sudoe Programme (European Regional Development Fund); and by the French government program Investissements d'Avenir via RobotEx Equipment of Excellence (ANR-10-EQPX-44) and IMobS3 Laboratory of Excellence (ANR-10-LABX-16-01).

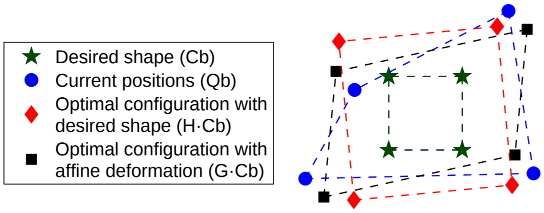
Desired shape (Cb)
Current positions (Qb)
Optimal configuration with desired shape (H·Cb)
Optimal configuration with affine deformation (G·Cb)

Fig. 2. Illustration of shapes used by our shape controller. For every one of the four shapes shown, the robot identities are 1-2-3-4, counted clockwise starting from the robot in top-right. Notice that the affine deformation is a stretching and shearing of the desired shape.

configuration with the desired shape. To explain the affine term, we refer to the concept of *deformation modes* (see, e.g., [6]). These modes encode ways in which real objects deform. Specifically, an affine deformation represents stretch and shear deformation modes. Figure 2 illustrates the action of the affine deformation. The specific action performed by the affine term is to move the robots towards the optimal configuration with affine deformation.

Therefore, the shape controller moves the team towards the desired shape (thanks to the pure deformation term) while maintaining plausible stretch and shear deformations along the way (thanks to the affine deformation term), instead of other unpredictable deformations which might damage the object being manipulated. The affine term is particularly useful when the deformation is large: in that case, it ensures that the shape transition towards the desired shape is done while respecting plausible deformations. In effect, the shape controller regulates the team shape by controlling the deformation measure variable. Aside from this, our approach also allows us to control the team's absolute state in the workspace: we do so by regulating three further variables: centroid, rotation, and size. Altogether, this gives four controlled variables. We call the approach we propose to control all of them a *full team controller*. Importantly, this full team controller consists of *fully decoupled* controllers for the four variables. By decoupled we mean that we can control the value of each of the variables without modifying the values of the others. This has obvious advantages as it enables a conflict-free and very flexible control design. Additional advantages of our control approach are that it requires few resources, and it is robust to perturbation (i.e., non-ideal motions). We focus our study on the case where the robot positions lie in 2D, and we also show the application of our approach in 3D.

Our contribution is relevant both theoretically and practically. Theoretically, our approach represents a new perspective relative to common formation control works, and we provide thorough formal guarantees of the performance. Such guarantees are not usual in, e.g., AI approaches based on neural networks. Practically, we identify several specific application scenarios where our approach can be directly useful. We discuss these scenarios next.

C. Example application scenarios

We identify three example target scenarios of multirobot manipulation of deformable objects: `(S1)` Manipulation of highly deformable objects, e.g., transport of fabrics [7], [8], or handling of a belt-like or a planar rubber object [9]. In these tasks, controlling the deformation of the robot positions can avoid damaging the object during reshaping and resizing (e.g., stretching). `(S2)` Manipulation with many robots [10] which are, e.g., deployed around the contour of a deformable object. Here the object's shape is determined directly by the robot team's shape, which our method can control suitably. `(S3)` Preservation of the shape formed by the fingertips during in-hand manipulation [11], to, e.g., maintain a stable object grasp or exert a given deformation. In these scenarios, the main usefulness of our approach is that it still allows one to guide the motions with a concept of deformation if information of contact forces or the object's shape is not available. The experimental evaluation we present illustrates these scenarios and the method's robustness to perturbation.

D. Comparison with preliminary paper

A preliminary version of the method we propose was presented in the paper [12]. Here, we extend that paper substantially. The following contents are new relative to [12]: *(i)* the affine control term, *(ii)* the decoupled controllers for all variables, *(iii)* a reformulation of the controller in a compact matrix form that facilitates interpretation and analysis, *(iv)* an extensive experimental validation which includes diverse scenarios of application, comparisons with methods in the literature and experiments with a real robotic hand, and *(v)* a discussion of the implementation of the approach in 3D.

E. Outline of the article

The contents of this article are structured as follows. In Section II we review the related work. Section III presents the definition of the problem addressed. In Section IV we describe the proposed shape controller. In Section V we describe in detail the full controller we propose, and discuss further the system's design. Section VI presents our experimental results. In Section VII we provide a concluding discussion.

II. Related work

A. Multirobot manipulation of deformable objects

Manipulation of deformable objects with a multirobot system is an increasingly popular topic [3]. In particular, recent works have addressed the transport of highly deformable objects with a team of robots [7], [8]. These works exploit distance constraints but do not use a global concept of deformation, as we propose here. In [13], [14], an aerial robot team is controlled using a deformation model of the specific payload –a flexible ring and a fabric, respectively– the team transports; in contrast, we give a general deformation-based coordination framework, and provide formal analysis of it. A different task with deformable objects is actively controlling the object's shape/deformation (i.e., *deformation control*). This typically requires multiple robots. Several works exist, with robots moving in 2D [5], [10], [15] and 3D [9], [11], [16], [17] workspaces. Generally they do not use a specific mechanism of coordination among robots; the coordination emerges implicitly from the collective task. Although here we do not address the precise control of the shape of an object, the proposed approach can be helpful in these tasks, especially when the capability to sense the shape of the object is limited. This is because we coordinate the robots based explicitly on minimizing deformation, which can avoid damaging the object. For this purpose we use global information of the team. What this means is the control law for each robot is computed using the relative positions of *all* the robots. The use of global information is known to facilitate tightly coordinated multirobot motions [18]–[20]; we note that such tight coordination is particularly critical in collaborative manipulation scenarios as the contact with an object poses additional risks and constraints.

B. Formation control

Perception, manipulation or navigation tasks that are fundamental in a range of key applications can be carried out by a team of multiple robots moving in a suitably coordinated way [4], [21]. The problem we address here (namely, keeping the multirobot team close to a prescribed geometric configuration) is often tackled using *formation control*, with well-established existing solutions based on controlling, e.g., relative positions [18], [22]–[25], distances [26]–[28], or angles/bearings [29], [30]. The problem we address features prominently, e.g., in works dealing with multi-robot transport of (quasi-)rigid objects; in these, the control is guided by the measured robot-object forces [31], and in some cases [20], [32], [33] formation control is used. In contrast with standard formation control approaches, typically based on pairwise robot interactions, we approach the addressed coordination problem using a measure of team deformation. This creates closely coordinated motions and adapts well to scenarios of deformable object manipulation, because in these deformation is a directly meaningful concept.

Since we control not just the shape but also the absolute configuration of the team, our work is closely connected with the problem of *formation maneuvering*. The method proposed in [22] addresses this problem and takes inertia into account. In particular, this method employs passive decomposition to decouple the dynamics of the team's shape from the dynamics of its centroid. In [27] distance mismatches are exploited to control the non-shape formation parameters (i.e., centroid, rotation and size) except the size one. Predefined maneuvers for formation tracking are proposed in [34] while other approaches use leader robots [35], [36], without focus on decoupling. Our approach differs in that we control all non-shape variables (not just some of them), do so in a decoupled manner, and our shape control strategy is explicitly based on deformation.

C. Robotics/AI and control systems engineering

Finally, it is interesting to position our article in a broader context within the systems engineering literature: doing so can help to find useful interdisciplinary connections, and to enlarge the scope of our work. As said above, the multirobot system we consider is an embodied AI system whose shape we control. It can therefore be considered a deformable robotic system. This fact connects our work with the field of soft robotics [37], which is growing rapidly in recent years. In particular, since we propose a control loop based on deformation, our work can fit the definition of soft controller proposed in [38]. Furthermore, our deformation-based formulation can have interest in other feedback control problems beyond the specific robotic problem we treat, as applications of feedback control are wide-ranging (e.g., from insulin delivery [39] to smart grid load management [40]).

III. Problem definition

A. Preliminaries

$\mathbf{I}_N$ denotes the $N \times N$ identity matrix. $\mathbf{1}_N$ notates a column vector of N ones. We define $\mathbf{S} = [(0,1)^T, (-1,0)^T]$, i.e., for a matrix $\mathbf{A} \in \mathbb{R}^{2\times N}$, $\mathbf{SA}$ is a counterclockwise rotation of $\pi/2$ radians of the column vectors of $\mathbf{A}$. Also, we notate as $\mathbf{A}^+ = \mathbf{A}^T(\mathbf{A}\mathbf{A}^T)^{-1}$ the $N \times 2$ pseudoinverse of $\mathbf{A}$, which satisfies $\mathbf{A}\mathbf{A}^+ = \mathbf{I}_2$. We define a centering matrix $\mathbf{K}_b = \mathbf{I}_N - \frac{1}{N}\mathbf{1}_N\mathbf{1}_N^T$, which is symmetric ($\mathbf{K}_b^T = \mathbf{K}_b$), idempotent ($\mathbf{K}_b^2 = \mathbf{K}_b$), and satisfies $\mathbf{K}_b\mathbf{1}_N = \mathbf{0}$. $||\cdot||$, $||\cdot||_F$ denote Euclidean and Frobenius norm, respectively. $tr(\cdot)$ denotes the trace operator. We generally do not notate time dependence. All proofs are given in the Appendix.

B. Problem statement

Consider a team of N robots in $\mathbb{R}^2$. The problem we address is to bring the robot positions towards a given configuration in the workspace while suitably controlling their shape. By *shape* we refer to the information about the set of positions that is left after translation, rotation, and scaling have been disregarded from consideration [41]. Accordingly, we use the term *non-shape variables* to refer to centroid, rotation and size of the set. The considered setup is illustrated in Fig. 1.

We define, then, a prescribed desired shape, encoded as a position $\mathbf{c}_i \in \mathbb{R}^2$ for each robot i. Moreover the current robot position at time t is denoted as $\mathbf{q}_i(t) \in \mathbb{R}^2$. Both $\mathbf{c}_i$ and $\mathbf{q}_i$ are expressed in the system's world frame. We represent the full sets of positions by stacking these column vectors, as: $\mathbf{Q} = [\mathbf{q}_1, \mathbf{q}_2, ..., \mathbf{q}_N]$, $\mathbf{C} = [\mathbf{c}_1, \mathbf{c}_2, ..., \mathbf{c}_N]$. $\mathbf{Q}$ and $\mathbf{C}$ are of size $2 \times N$. We define the centroid of the current positions as $\mathbf{q}_0 = \frac{1}{N}\mathbf{Q} \cdot \mathbf{1}_N$. In stacked matrix form this is $\mathbf{Q}_0 = \mathbf{q}_0\mathbf{1}_N^T$. Analogously $\mathbf{C}_0 = \frac{1}{N}\mathbf{C} \cdot \mathbf{1}_N\mathbf{1}_N^T$. We then define the matrices of positions with zero centroid as:

$$\mathbf{Q}_b = \mathbf{Q} - \mathbf{Q}_0 = \mathbf{Q}\mathbf{K}_b, \;\; \mathbf{C}_b = \mathbf{C} - \mathbf{C}_0 = \mathbf{C}\mathbf{K}_b. \quad (1)$$

In addition, we consider a desired absolute configuration, $\mathbf{Q}_T$ $\in \mathbb{R}^{2\times N}$. We parameterize it as follows:

$$\mathbf{Q}_T = s_d\mathbf{R}_d(\theta_d)\mathbf{C}_b + \mathbf{Q}_{0d}, \quad (2)$$

with $\mathbf{Q}_{0d} = \mathbf{q}_{0d}\mathbf{1}_N^T \in \mathbb{R}^{2\times N}$, $\mathbf{R}_d(\theta_d) \in SO(2)$ and $s_d \in \mathbb{R}_{>0}$. Notice that $\mathbf{Q}_T$ has the desired shape, since it consists in a rotation and uniform scaling of $\mathbf{C}_b$, plus a translation. The non-shape variables of $\mathbf{Q}_T$ are encoded by $\mathbf{q}_{0d} \in \mathbb{R}^2$ (centroid), $\mathbf{R}_d(\theta_d)$ (rotation) and s_d (size).

We consider a single-integrator motion model: $\dot{\mathbf{q}}_i = \mathbf{u}_i$, where $\mathbf{u}_i \in \mathbb{R}^2$ is robot i's control input. Thus: $\dot{\mathbf{Q}} = \mathbf{U}$, where $\mathbf{U} = [\mathbf{u}_1, \mathbf{u}_2, ..., \mathbf{u}_N]$ is the team's control input. We name our control laws using $\mathbf{U}$ with identifying subscripts.

Control goals: The desired shape is achieved if $\mathbf{Q}$ is equal to $\mathbf{C}$ up to a rotation, uniform scaling, and translation. Using $\mathbf{Q}_b$ and $\mathbf{C}_b$, the translation element can be removed. Then, we formally define our **shape control goal** by the condition:

$$\mathbf{Q}_b = s\mathbf{R}(\theta)\mathbf{C}_b, \tag{3}$$

for a generic $s \in \mathbb{R}_{\geq 0}$ and a generic $\mathbf{R}(\theta) \in SO(2)$.

In addition to the shape control goal (3), we also define our **full control goal**, by the following condition:

$$\mathbf{Q} = \mathbf{Q}_T. \tag{4}$$

An additional specification in our problem definition is that the multirobot control system to be designed needs to be feasible in terms of resource consumption. Moreover, it is desirable for the system to satisfy other nonfunctional criteria such as robustness to perturbation.

IV. Shape controller

Our strategy to achieve the shape control goal (3) is based on the cost function:

$$\gamma = \frac{1}{2}||\mathbf{Q}_b - \mathbf{H}\mathbf{C}_b||_F^2. \tag{5}$$

$\mathbf{H} \in \mathbb{R}^{2\times 2}$ has the structure of a similarity, i.e.,:

$$\mathbf{H} = \begin{bmatrix} h_1 & -h_2 \\ h_2 & h_1 \end{bmatrix}, \tag{6}$$

and the values of its components are:

$$h_1 = \frac{tr(\mathbf{Q}_b\mathbf{C}_b^T)}{c_s},\ \ h_2 = \frac{tr(\mathbf{Q}_b(\mathbf{S}\mathbf{C}_b)^T)}{c_s}, \tag{7}$$

with $c_s = tr(\mathbf{C}_b\mathbf{C}_b^T) = ||\mathbf{C}_b||_F^2$. Any real transformation having the structure (6) performs uniform scaling and rotation when acting on $\mathbf{C}_b$. Within the set of such transformations, $\mathbf{H}$ is the one that aligns with least-squares error $\mathbf{C}_b$ with $\mathbf{Q}_b$; i.e., $\mathbf{H}$ is the solution of a Procrustes optimal alignment problem, found by differentiation from (5). Note that we can express $\mathbf{H} = s_h\mathbf{R}_h$, with $s_h = \sqrt{h_1^2 + h_2^2} = tr(\mathbf{Q}_b^T\mathbf{R}_h\mathbf{C}_b)/c_s$ being a scaling factor and $\mathbf{R}_h(\theta_h)$ a counterclockwise rotation by angle $\theta_h = atan2(h_2, h_1)$. Note that, as will be further explained in Remark 1, we disregard the case $s_h = h_1 = h_2 = 0$. We do so because (i) this case corresponds to configurations with zero measure, i.e., not relevant in practice since an infinitesimal perturbation takes the system out of them and (ii) we will show how to specifically control s_h, which means this variable cannot go to zero.

One notices directly in (5) that γ is a certain measure of error between the current $\{\mathbf{q}_i\}$ and desired $\{\mathbf{c}_i\}$ positions. Since $\mathbf{H}$ is *optimal*, γ abstracts out the non-shape parameters, and hence characterizes the difference *in shape* between the two sets. From the viewpoint of the set $\{\mathbf{c}_i\}$, this difference expresses how much the set $\{\mathbf{q}_i\}$ is deformed relative to it. Therefore, γ measures the deformation of the current robot positions with respect to their desired shape. Note in the next lemma that γ captures exactly our shape control goal.

Lemma 1: (3) is satisfied if and only if $\gamma = 0$. $\diamond$

We compute next the gradient of γ with respect to the robot positions. Note that the gradient of γ with respect to $\mathbf{H}$ is zero, as $\mathbf{H}$ is a minimizer of γ. Hence:

$$\nabla_{\mathbf{Q}}\gamma = (\mathbf{Q}_b - \mathbf{H}\mathbf{C}_b)\mathbf{K}_b^T = \mathbf{Q}_b - \mathbf{H}\mathbf{C}_b. \tag{8}$$

Pure deformation control term: We define a control term for $\mathbf{Q}$ as the negative gradient of γ. Using (8):

$$\mathbf{U}_H = -\nabla_{\mathbf{Q}}\gamma = \mathbf{H}\mathbf{C}_b - \mathbf{Q}_b. \tag{9}$$

With this control term the team executes, at every time instant, the motion that optimally reduces the deformation measure γ. We highlight that the similarity $\mathbf{H}$ is not constant; indeed, $\mathbf{H}(t)$ is computed from the positions $\mathbf{Q}(t)$.

Affine deformation control term: Using (9) to control a large deformation, one has no control over how the shape will evolve from initial to desired, and thus during the transition the team may acquire shapes that do not correspond with suitable deformations. To address this issue, we propose to augment the controller (9) with an *affine deformation control term*. This term is based on the affine transformation of matrix $\mathbf{C}$ that aligns it optimally (in least-squares sense) with $\mathbf{Q}$. Finding this transformation is a well-known problem in linear regression. Its solution is computed by making the two centroids coincide, and applying on $\mathbf{C}_b$ the affine transformation $\mathbf{G} \in \mathbb{R}^{2\times 2}$ defined as:

$$\mathbf{G} = \mathbf{Q}_b\mathbf{C}_b^+. \tag{10}$$

Notice that $\mathbf{C}_b^+$ is a *constant* matrix (as $\mathbf{C}_b$ is fixed). The affine deformation control can be used for any desired geometry of robot positions that is not a straight line (because then $\mathbf{C}_b$ is not full rank). Note that the pure deformation control (9) can be used for any nontrivial geometry of $\mathbf{C}_b$.

Then, we define the affine deformation control term as:

$$\mathbf{U}_G = \mathbf{G}\mathbf{C}_b - \mathbf{Q}_b. \tag{11}$$

An affine transformation encodes stretch and shear (Fig. 2), which are plausible deformation modes in which real-world objects deform. Although we do not control the deformation of an object, rather the deformation of the team that manipulates it, clearly the two deformations are closely related; therefore it is interesting to restrict the team's deformation to these plausible modes. This is what $\mathbf{U}_G$ does, as it continuously moves the team towards the shape ($\mathbf{G}\mathbf{C}_b$) that corresponds with the optimal affine deformation of the desired one.

Shape controller: We propose, based on (9) and (11), the following controller aimed at fulfilling (3):

$$\mathbf{U}_\gamma = \alpha_H\mathbf{U}_H + \alpha_G\mathbf{U}_G, \tag{12}$$

where $\alpha_H > 0$ and $\alpha_G \geq 0$ are control weights. These weights can be chosen adaptively; a larger α_G can be especially useful when the deformation is large, while a larger α_H can increase convergence rate. A notable property of the shape controller (12) is that it is a *linear* controller of $\mathbf{Q}$; linear control has well-known advantages in practice. The idea of this controller is to make the robots eventually converge to the desired shape suitably reducing γ (due to $\mathbf{U}_H$), while staying close to an affine deformation along the way (due to $\mathbf{U}_G$). For this, the two control terms need to be non-conflicting. We will show two facts which demonstrate that this is the case. First, the following property holds true.

Proposition 1: s_h, $\mathbf{R}_h$ and $\mathbf{q}_0$ are all invariant under $\mathbf{U}_\gamma$. $\diamond$
Second, $\mathbf{U}_G$ can never increase γ. We state next a lemma that will be used later on to show this fact.

Lemma 2: Given matrices $\mathbf{A}$, $\mathbf{B} \in \mathbb{R}^{2\times N}$, $\mathbf{B}$ being full-rank, and $\mathbf{H}_w = s_w\mathbf{R}_w$, $s_w \in \mathbb{R}^{\geq 0}$, $\mathbf{R}_w \in SO(2)$, it holds that:

$$||\mathbf{A} - \mathbf{H}_w\mathbf{B}||_F^2 = ||\mathbf{A} - \mathbf{M}\mathbf{B}||_F^2 + ||\mathbf{M}\mathbf{B} - \mathbf{H}_w\mathbf{B}||_F^2, \quad (13)$$

where $\mathbf{M} = \mathbf{A}\mathbf{B}^+$. $\diamond$
Notice too that as (12) depends on $\mathbf{Q}_b$ (not $\mathbf{Q}$), it may be computed with relative (not absolute) position measurements.

V. Full team controller

Let us now address how to coordinate the robots to achieve the full control goal (4). We define four variables that capture the state of the team: γ, $\mathbf{q}_0$ θ_h and s_h. Our control goal is then to bring γ to 0, $\mathbf{q}_0$ to $\mathbf{q}_{0d}$, θ_h to θ_d, and s_h to s_d. To do so, we define a set of four error variables:

$$\begin{aligned} &\gamma \in \mathbb{R}, \quad \mathbf{e}_0 = \mathbf{q}_0 - \mathbf{q}_{0d} \in \mathbb{R}^2, \\ &e_\theta = \theta_h - \theta_d \in \mathbb{R}, \quad e_s = s_h - s_d \in \mathbb{R}. \end{aligned} \quad (14)$$

We already have a controller for γ (12). We next propose controllers for the three non-shape variables.

A. Decoupled full team control

The proposed control terms are as follows:

$$\mathbf{U}_0 = -\alpha_0\mathbf{e}_0\mathbf{1}_N^T, \quad \alpha_0 > 0, \quad (15)$$
$$\mathbf{U}_\theta = -\alpha_\theta e_\theta \mathbf{S}\mathbf{H}\mathbf{C}_b, \quad \alpha_\theta > 0, \quad (16)$$
$$\mathbf{U}_s = -\alpha_s e_s (1/s_h)\mathbf{H}\mathbf{C}_b, \quad \alpha_s > 0. \quad (17)$$

Here, $\mathbf{U}_0$ displaces the team as a whole towards centroid $\mathbf{q}_{0d}$. Notice that $\mathbf{U}_\theta$ and $\mathbf{U}_s$ are *nonlinear* controllers of $\mathbf{Q}$, and that they are defined by a rotation and scaling, respectively, of the configuration $\mathbf{H}\mathbf{C}_b$. This is the destination configuration that $\mathbf{Q}_b$ is moving towards, as can be seen by the expression of the pure deformation control term (9).

Remark 1: We assume $s_h > 0$ at time zero and, for convenience and without loss of generality, $\theta_d = 0$. Note one can always rotate $\mathbf{C}_b$ (such rotation changes no aspect of our coordination approach) to make $\theta_d = 0$. We define by convention $\theta_h(t = 0)$ in the range $(-\pi, \pi]$. Then, note that $s_h(t) > 0$ and $\theta_h(t) \in (-\pi, \pi]$ $\forall t > 0$ because, as shown later, $\mathbf{U}_s$ and $\mathbf{U}_\theta$ control these variables in a fully decoupled way. These assumptions ensure $\theta_h(t)$ will be smooth. $\square$
Let us call a controller for a variable (γ, $\mathbf{q}_0$, θ_h, or s_h) *decoupled* if it does not change the values of the other variables. We can give our result on decoupling next.

Proposition 2: $\mathbf{U}_\gamma$, $\mathbf{U}_0$, $\mathbf{U}_\theta$, and $\mathbf{U}_s$ are decoupled controllers for γ, $\mathbf{q}_0$, θ_h, s_h respectively. $\diamond$
We then propose the following **full controller**:

$$\mathbf{U}_f = \mathbf{U}_\gamma + \mathbf{U}_0 + \mathbf{U}_\theta + \mathbf{U}_s. \quad (18)$$

The following is the main convergence result of this article.

Theorem 1: The full controller $\mathbf{U}_f$ (18) brings the team to the target configuration $\mathbf{Q}_T = s_d\mathbf{R}_d(\theta_d)\mathbf{C}_b + \mathbf{Q}_{0d}$. Moreover the dynamics follows an exponential decoupled decrease for $\mathbf{e}_0$, e_θ, and e_s; and, if $\alpha_G = 0$, also for γ. $\diamond$

Remark 2: We stress that what makes the decoupling possible is that we have defined a measure (γ) that captures *purely* the deformation. Important advantages of the decoupling are: (i) It preserves the properties of the shape controller $\mathbf{U}_\gamma$ (12), while fulfilling goal (4) with $\mathbf{U}_f$ (18); (ii) It can facilitate extending the controller to *time-varying* references $\mathbf{C}(t)$, $\mathbf{q}_{0d}(t)$, $\theta_d(t)$, $s_d(t)$, which could be tracked in a decoupled manner; (iii) It directly allows *partial* control, leaving some parameters invariant, e.g.: rotating and resizing the team without translation; (iv) Desired behaviors for the variables can be designed by appropriately selecting the weights α; (v) Controller convergence is guaranteed for any weights α. Note that [12] proposed control laws (*"variants"*) that could control either rotation or scaling while preserving the other variable, or used a leader robot. However full decoupling was not guaranteed (e.g., γ could increase) and stability was not analyzed. Here the control is fully decoupled. $\square$

B. Alternative controls for rotation and size

Rotation and resizing of the current shape: An alternative strategy is to rotate and resize the current team shape, $\mathbf{Q}(t)$. Although this does not ensure full decoupling from γ, it can be effective in practice. We propose the controls:

$$\mathbf{U}_{\theta_q} = -\alpha_{\theta_q} e_\theta \mathbf{S}\mathbf{Q}_b, \quad \alpha_{\theta_q} > 0, \quad (19)$$
$$\mathbf{U}_{s_q} = -\alpha_{s_q} e_s \mathbf{Q}_b, \quad \alpha_{s_q} > 0. \quad (20)$$

Integrated rotation and size control: This strategy is designed to regulate *simultaneously* the rotation and size parameters. A motivation for this compact strategy is that it can be implemented in both the 2D case and the 3D case (discussed in Section V-D). We propose a control term based on reaching a desired similarity $\mathbf{H}_d = s_d\mathbf{R}_d$:

$$\mathbf{U}_{H_d} = \alpha_{H_d}(\mathbf{H}_d\mathbf{C}_b - \mathbf{Q}_b), \quad \alpha_{H_d} > 0. \quad (21)$$

This term is not decoupled from γ but it gives a suitable and intuitive way to make $\mathbf{H}$ gradually transition towards $\mathbf{H}_d$.

C. Inter-robot distances

A controller based on team deformation can help to maintain suitable distances between *every* pair of robots. Exploiting the interesting fact that γ and e_s can be controlled to converge monotonically, we provide the following explicit bounds which build on Prop. 3 in [12] and can be proven in a straightforward manner using the mentioned monotonic convergence.

Proposition 3: Define $e = 2\sqrt{N\gamma(t=0)}$; using controller (18), the distance d_{ij} between robots i and j remains bounded, $\forall i, j$, as: $min(s_h(0), s_d)||\mathbf{c}_i - \mathbf{c}_j|| - e \leq d_{ij} \leq max(s_h(0), s_d)||\mathbf{c}_i - \mathbf{c}_j|| + e$. $\diamond$

D. The 3D case

In our study we considered that the robot positions lie on a plane. Note that this is the case in several deformable object transport and manipulation scenarios, even in cases where the manipulated object lies in 3D space [5], [7], [8], [10], [14], [15]. This fact highlights the applicability of the approach we proposed and studied. If robots move in 3D, the proposed approach can still be implemented, as follows. We can encode the positions in matrices $\mathbf{Q}$ (and $\mathbf{Q}_b$), $\mathbf{C}$ (and $\mathbf{C}_b$), now of size $3 \times N$. We first compute from these matrices the optimal rotation $\mathbf{R}_h \in SO(3)$. This can be done via the well-known Kabsch algorithm that solves the orthogonal Procrustes problem. Then, we can use this rotation matrix to find the optimal scaling, analogously to the 2D case (see Section IV), as $s_h = tr(\mathbf{Q}_b^T \mathbf{R}_h \mathbf{C}_b)/c_s$. This gives the similarity $\mathbf{H} = s_h \mathbf{R}_h$, with which we can implement the pure deformation term (9). The affine deformation term can be computed using (11), and (10). The non-shape parameters can be controlled using (15), (17), (20), (21). The Kabsch algorithm uses Singular Value Decomposition. It is therefore not analytic and hence this 3D solution cannot be analyzed as the 2D one. Still, we show in the tests in Sect. VI that the controller implementation in 3D is functional.

E. System deployment and nonfunctional criteria

We propose a multirobot coordination method based on global information, instead of one based on partial information. What we mean precisely by global information is that the proposed method uses the relative positions of *all* the robots. Our motivations for this choice are: (i) Global information is known to be a suitable choice when tight coordination is needed [18]–[20]; in particular, for the problem we address, a distributed controller using partial information cannot produce instantaneous motions optimizing deformation globally. (ii) In manipulation tasks the number of robots is often small, so scalability is not critical. (iii) Certain manipulation systems are intrinsically centralized (e.g., robotic hand fingers).

After having addressed the functional aspects (i.e., the controller design) of our system in previous sections, we discuss several relevant nonfunctional aspects next.

Architecture: The proposed multirobot controller is not tied to a specific architecture, and several choices are possible. We illustrate two possible architectures in Fig. 3. A centralized architecture with a central node (or leading robot) handling all the data processing is more efficient and less complex overall. It can adapt well to scenarios where the system is intrinsically centralized. An example is a robotic hand, where the hand controller would be the central node for the fingers (robots). A distributed architecture, on the other hand, is more resilient [21], [42]–[44] as each robot can take the role of leader [1]. It can be more appropriate when using mobile robots, each having a high degree of autonomy. Note that a distributed architecture is a suitable possible choice for our controller because the controller does not have high requirements in terms of data volumes and computing power.

Mission specification: It consists of the tuple $\{\mathbf{C}, \mathbf{q}_{0d}, \theta_d, s_d\}$, and the control weights α. How this information is determined will depend on the specific application scenario. From the knowledge of the robot-object contact points, one can link the state of the object with the positions of the robots, and thus define $\mathbf{C}$. As examples, for a transport task $\mathbf{C}$ can be the contact positions when the object is at rest, and for laying a cloth on a table, $\mathbf{C}$ can be computed from the table's shape and dimensions.

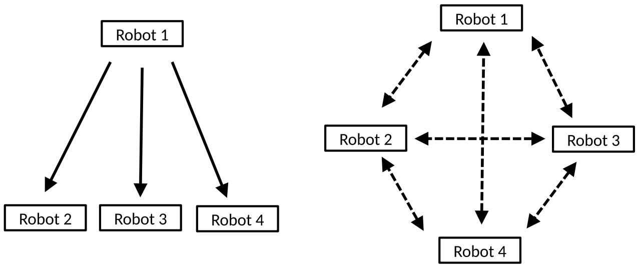

Fig. 3. Possible system architectures exemplified with 4 robots. Arrows represent communication links, where a dashed line indicates the link may or may not be present. Left: Centralized system. Right: Distributed system.

Sensing and communications: The robots need to know the positions of the other robots. For this they can self-localize (using, e.g., a GPS-like sensor) and communicate their positions, or use other sensors (e.g., vision) to perceive the other robots' relative positions. The identity of the specific robot associated to each position measurement needs to be known too. These identities can be obtained either by sensing or communication. Different message transmission schemes (point-to-point, broadcast, multi-hop) are possible in the system. Generally the data volume to be exchanged per cycle (i.e., the robots' position measurements) scales linearly with the number of robots. Additionally, in the distributed architecture, all robots need to receive (via communications, and prior to execution) the mission specification.

Computation: The controller (18) is very light to compute because it is an analytical expression based on standard mathematical operations, and does not rely on executing iterative algorithms or computing optimizations.

Memory: The storage needed increases linearly with the number of robots. Large data structures are not needed as the data volume required per robot is light (only a position vector). The mission specification is also lightweight.

Robustness to perturbation: Coordination controllers that follow the negative gradient of a cost function (such as our shape controller) are robust to perturbations in the motion directions of up to 90 degrees. This is because under such perturbations the actual motion direction is still in the same half-plane as the negative gradient vector, and thus that motion is still reducing the cost function [25], [45]. We illustrate this interesting robustness property in our experiments.

VI. Experiments

Our goals in this section are to corroborate experimentally our theoretical findings and to validate the usefulness of the proposed approach. For these purposes we test the approach in several scenarios and different conditions, which include perturbations. As the motion model we consider is single-integrator, the robots are simulated as point masses in Secs. VI-A, VI-B and VI-C. This allows us to provide a general evaluation, independent of specific underlying robot models. Still, as indicated at the end of Sec. V-E, our approach is robust if additional kinematic constraints cause the motion not to follow the single-integrator model; this is illustrated in Sec. VI-D where we apply the approach to control the fingertips of a robotic hand (both simulated and real).

We include comparisons, mainly with formation controller [23] which is linear (like our shape controller) and has been used for formation preservation during manipulation in, e.g., [33]. We implement it assuming a full formation graph (i.e., every robot knows the position of all others), which is the case considered in our work. We used Matlab, except for the tests with the robotic hand, for which we used ROS and Python. A **video** illustrating tests for the different scenarios is attached and accessible at `https://youtu.be/WD98yZZJUuM`.

A. *Team coordination with large deformation*

We simulated six robots moving towards a regular-hexagonal desired configuration. We considered a large initial deformation. This example intends to specifically illustrate the usefulness of our affine control term: this term is particularly important and helpful when the deformation to be controlled is large. We chose $\alpha_G = 10\alpha_H$. We used controller $\mathbf{U}_f$ (18). The first row of Fig. 4 shows that this controller produces a shape transition where the robots preserve a plausible affine deformation. In contrast, [23] lacks this control of transition. This can produce unsuitable motions: e.g., the polygon formed by the robot positions becomes self-intersecting (see plot on second row, second column). This is clearly not appropriate if the robots are transporting a deformable object (scenario (`S1`)). Implementing [23] with a distributed, incomplete graph (e.g., each robot having only two neighbors) may help to avoid such self-intersections; however, the downside is that the motions are less efficient and, more importantly, the team can shrink/stretch/deform without control during its transition to the desired shape. We also test, and illustrate in Fig. 4, a strategy based on the nonlinear rigidity-based approach [26]. Again, the shape transition is worse than with our approach. Importantly, [26] is stable only *locally*, and for large deformations it may converge to undesired equilibrium configurations (even with a full formation graph). In contrast, our controller is stable *globally* and converges to the desired shape from any initial condition. Finally, the last row in Fig. 4 shows the result with our method when used without the affine term (i.e., choosing $\alpha_G = 0$). It can be seen that the motions are similar to those obtained with [26]. This illustrates the importance of using the affine term when the deformation is large.

B. *Handling an elastic object with perturbations*

This example also illustrates scenario (`S1`). Consider an elastic sheet and a task of transporting it or making its perimeter fully contain a given area. In this case: *(i)* The sheet's exact shape does not need to be controlled specifically, but one needs to control its deformation to avoid damaging it and *(ii)* as the sheet is highly deformable and has very low stiffness, its deformation is determined by the team deformation. Hence our approach is useful as it controls the team deformation, and does not require knowing the sheet's shape.

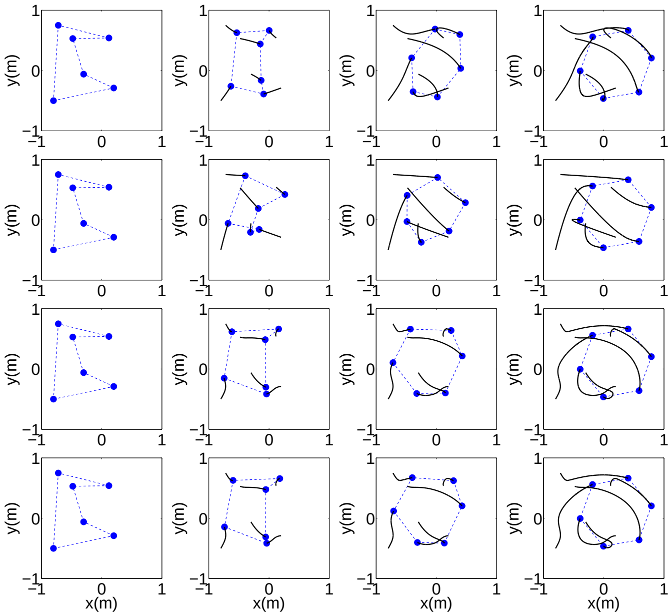

Fig. 4. Six-robot simulation. Left to right: Four snapshots of team evolution for our approach (18) with $\alpha_G = 10\alpha_H$ (top row), [23] (second row), [26] (third row), and our approach (18) with $\alpha_G = 0$ (bottom row). The robots are shown as circles, their paths as solid black lines and the polygon formed by the robot positions is represented using dashed lines.

To simulate the elastic sheet we implemented the well-known approach Meshless Shape Matching (MSM) [6]. We used controller $\mathbf{U}_f$. We introduced perturbations to the model $\dot{\mathbf{q}}_i = \mathbf{u}_i$, to test robustness. Specifically we included: noise in $\mathbf{U}$ (which models both sensing and actuation error), actuation saturation (via a different maximum velocity limit for each robot) and non-identical control gains, to model the robots as heterogeneous. The task consisted in the team moving to two consecutive target configurations. Figure 5 shows that despite the strongly perturbed velocities the controller (18) can perform the task correctly and produces a suitable evolution of the object's shape. Again, we compare with [23], run under the same perturbations as our approach. [23] produced larger errors in some variables and more contorted paths.

To illustrate the object's behavior in the tests, we define γ_{obj} as the analog of γ but defined for the positions of the set of nodes representing the object's shape (instead of for the positions of the set of robots). We tested defining γ_{obj} with the standard optimal similarity transformation, and with a rotation (i.e., fixing $s_h = 1$), which yielded similar results. We ran 10 trials with varying saturation thresholds and noise, and averaged the results. The mean value of γ_{obj} was 56 % higher with [23] than with our approach. Note that we discarded all data below a certain small threshold value of γ_{obj}, to remove any effect of the convergence speed on these comparative results. Moreover, the maximum value of γ_{obj} was around 9 % higher with [23]. These results show that with our method

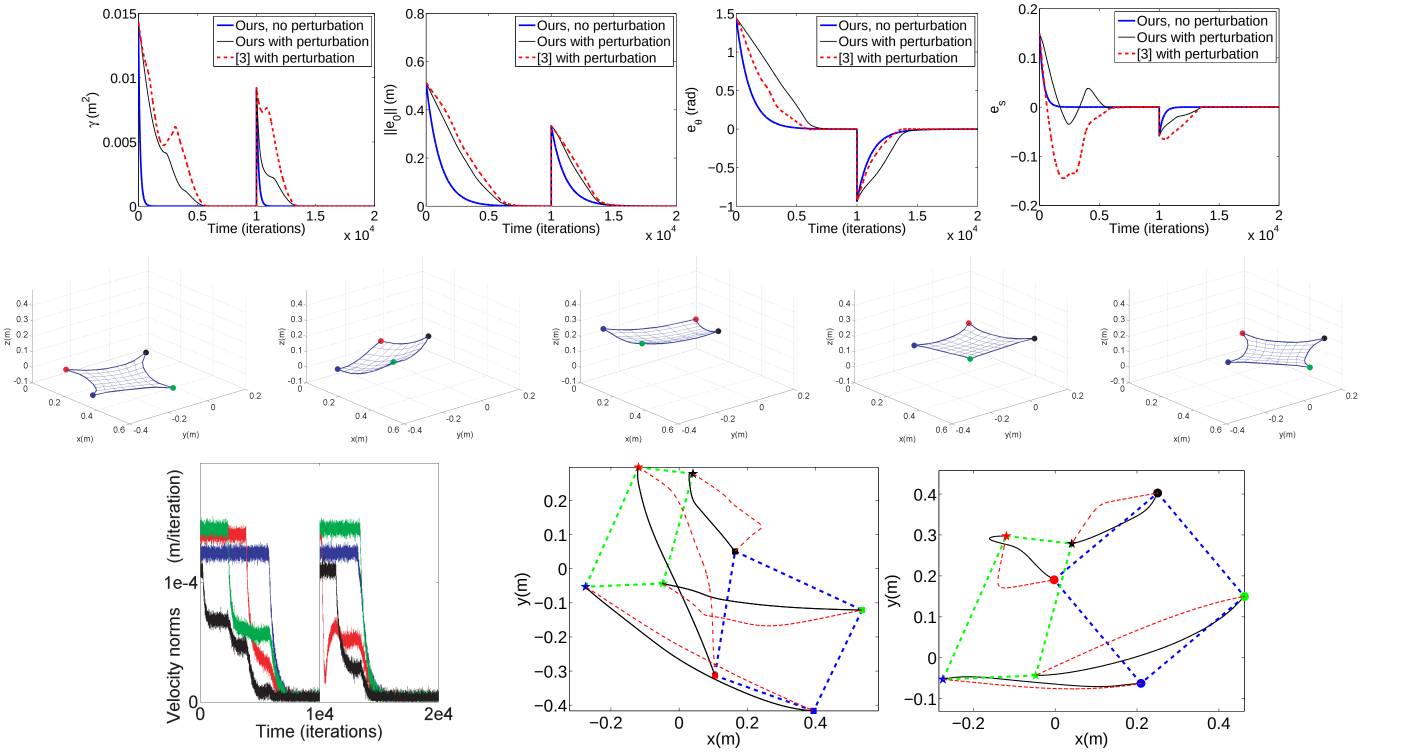

Fig. 5. Four-robot simulation. Top: time evolution of γ, $||\mathbf{e}_0||$, e_θ, e_s. Second row: snapshots of the object's shape; left to right: initial, intermediate, first target, intermediate, second target. Bottom, left to right: norms of robots' velocities, robot paths from initial configuration to first target, robot paths from first to second target. First target is marked with green dashed lines. Paths shown in red dashed lines for [23], in black solid lines for our method.

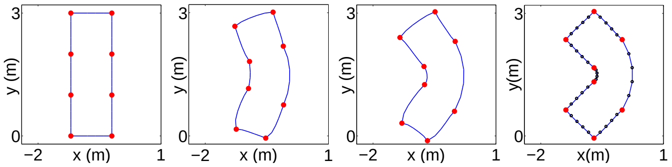

Fig. 6. Eight-robot simulation. Left to right: Four snapshots of system evolution. Robots as circles, object boundary as solid line, (in fourth plot) desired positions of object particles on the boundary as darker circles.

the shape of the object being manipulated evolved in a more uniform and efficient fashion.

C. Coordination for object shape control with many robots

We simulated eight robots manipulating an elastic, bar-shaped object lying in 2D and simulated with MSM. First, starting from the rest shape of the object, we moved manually the robots so that they bent it into a desired state. Then, we ran our controller starting from the rest shape, with $\mathbf{C}$ equal to the team's shape in the desired state. We used $\mathbf{U}_\gamma + \mathbf{U}_0 + \mathbf{U}_{H_d}$. We added motion disturbances similarly to the test in Section VI-B. The results in Fig. 6 attest that both the team and object deformations progressed appropriately during execution. This example illustrates scenario `(S2)`.

D. Robotic hand experiments

We applied our controller to coordinate the fingertip motions of a robotic hand –scenario `(S3)`–. We used the anthropomorphic Shadow Dexterous Hand. In all tests we implemented the controller in 3D (see Section V-D). The palm of the hand was fixed. As a joint velocity controller was not available for the hand we used, we employed position control. To control the position of every fingertip, we used inverse kinematics computed numerically via sequential quadratic programming. Fingertip control based on inverse kinematics is also employed, e.g., in [11]. Inverse kinematics for the robotic hand suffers from inaccuracies and oscillations. We created dead zones in the neighborhood of the Cartesian positions sent as commands to the fingertips: this eliminated oscillatory motions, at the cost of increasing the steady-state positioning error. We introduced saturation to the joint angle velocities, to avoid sudden fast motions and large differences between the motions of different fingers. Additionally the mechanical structure of the hand constrains finger mobility. Our goal was to show that the controller can still be functional in these challenging conditions.

Simulation: Figure 7 illustrates an example, visualized in RViz, of the fingertips moving towards a desired shape. We used $\mathbf{U}_\gamma + \mathbf{U}_0 + \mathbf{U}_{s_q}$. There were noticeable steady-state control errors but the motions remained stable. These errors were mainly due to the dead zones we had to introduce, as explained in the preceding paragraph. The coordination of the team is observable by the decrease of γ and the gradual and steady evolution of the shape formed by the fingertips.

Tests with real robotic hand: We installed the hand on a fixture and used a soft foam bar as an object to be manipulated. The goal was to show operation in contact with a deformable object further limiting finger mobility. The setup is shown in Fig. 7 along with results from a test. We used $\mathbf{U}_\gamma + \mathbf{U}_0 + \mathbf{U}_{H_d}$.

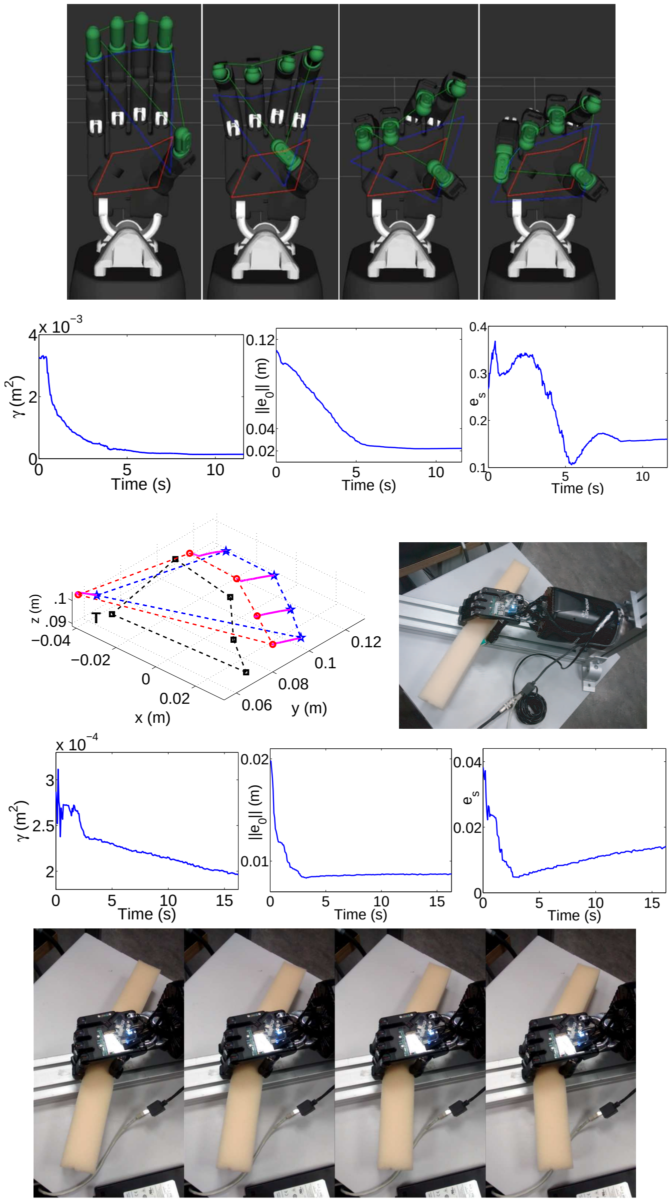

Fig. 7. Hand tests results. Top, left to right: initial, intermediate (two plots) and final shape for a test. Desired shape is the lowest, fixed red polygon. The commanded next fingertip positions are also shown as a changing blue polygon. Second row, left to right: evolution of γ, centroid and scaling errors for a simulation test. Third row, left: test with real hand, showing initial –stars– , final –circles– and desired –squares– fingertip positions. Fingertip paths in solid lines. Thumb marked as **T**. Third row, right: experimental setup. Fourth row, left to right: γ, centroid and scaling errors for real hand test. Bottom: four snapshots from initial (left) to final (right) configuration with the hand grasping and deforming the object under the action of the controller.

We verified that the controller maintained a stable behavior around the desired shape of the fingertips. The controller allowed the hand to grasp the deformable object, as shown in the snapshots in Fig. 7. Therefore, the proposed approach can control moderate displacements and is useful in this scenario mainly due to the difficulty of perceiving the state of the object. Still, note that we ignored aspects such as hand structural constraints or contact kinematics [46] which are very important especially for larger displacements.

VII. Discussion and Conclusion

We have presented a new approach to control a multirobot system in a coordinated manner based on the concept of deformation. We have highlighted the novelty of our formulation and its usefulness for scenarios of interest, particularly in problems related with the manipulation of deformable objects.

A relevant issue to discuss is the robot-object interaction. Our approach is geometric and does not consider object-robot contact forces. Still, we note that *(i)* we take deformation into account –geometrically– and this can help maintain suitable contact forces when there are no force sensors, and *(ii)* contact force measurements are not considered in several works on deformable object shape control [5], [16], [17], assembly [9], or transport [7].

Our method only controls the robots' positions (as in, e.g., [9]–[11], [14], [16], [32]), and not their orientations. End-effector orientation control can be necessary depending on the specific type of grasp and contact to, e.g., avoid exerting excessive torques on the object. Such orientation control can be added to the position controller we propose.

We conclude by proposing several directions for future work. An interesting extension is to address secondary objectives (e.g., control object shape, include contact constraints such as friction [46]...) exploiting the gradient-based character of the shape controller [45]. Control of higher-order dynamics (e.g., double-integrator model) is practically relevant and may be facilitated by the linearity of the shape controller. Tracking of a dynamically changing $\mathbf{Q}_T$ can benefit from the presented fully decoupled controls of the parameters we use to characterize it. Finally, for higher scalability and resilience to malfunctions, it is also of interest to execute some elements of the proposed approach in a distributed manner relying only on partial information and partial resources of the system.

Appendix

We first provide a list of expressions for the traces of products that we will use in the analysis. These identities can be deduced starting from $\mathbf{HC}_b = h_1\mathbf{C}_b + h_2\mathbf{SC}_b$, $\mathbf{SHC}_b = -h_2\mathbf{C}_b + h_1\mathbf{SC}_b$, and using straightforward manipulations.

$$tr(\mathbf{Q}_b\mathbf{C}_b^T) = tr(\mathbf{HC}_b\mathbf{C}_b^T) = h_1 c_s \tag{22}$$

$$tr(\mathbf{Q}_b(\mathbf{SC}_b)^T) = tr(\mathbf{HC}_b(\mathbf{SC}_b)^T) = h_2 c_s \tag{23}$$

$$tr(\mathbf{Q}_b(\mathbf{HC}_b)^T) = tr(\mathbf{HC}_b(\mathbf{HC}_b)^T) = (h_1^2 + h_2^2)c_s \tag{24}$$

$$tr(\mathbf{Q}_b(\mathbf{SHC}_b)^T) = tr(\mathbf{HC}_b(\mathbf{SHC}_b)^T) = 0. \tag{25}$$

Proof of Lemma 1: i) Suppose $\gamma = 0$. This implies $\mathbf{Q}_b = \mathbf{HC}_b$, and $\mathbf{H}$ has the form $\mathbf{H} = s_h\mathbf{R}_\mathbf{h}$. Hence, (3) is satisfied. *ii)* Assume now (3) holds, i.e., $\mathbf{Q}_b = s\mathbf{R}(\theta)\mathbf{C}_b$ where θ is the angle of the rotation. Let us verify what the transformation $\mathbf{H}$ (7) is for this pair $\mathbf{Q}_b$ and $\mathbf{C}_b$. We see that $h_1 = \frac{tr(\mathbf{Q}_b\mathbf{C}_b^T)}{tr(\mathbf{C}_b\mathbf{C}_b^T)} = s\frac{tr(\mathbf{RC}_b\mathbf{C}_b^T)}{tr(\mathbf{C}_b\mathbf{C}_b^T)} = s\cos(\theta)$, and similarly $h_2 = s\sin(\theta)$. So $\mathbf{H} = s\mathbf{R}$, and directly $\gamma = 0$.

Proof of Proposition 1: We will study $\dot{\mathbf{H}}$ starting with $\dot{h}_1$:

$$\dot{h}_1 = \frac{1}{c_s}\frac{d(tr(\mathbf{Q}_b\mathbf{C}_b^T))}{dt} = \frac{1}{c_s}tr(\frac{d(\mathbf{Q}_b\mathbf{C}_b^T)}{dt}). \tag{26}$$

We now consider the following expression in terms of two addends: $d(\mathbf{Q}_b\mathbf{C}_b^T)/dt = dh_{1G} + dh_{1H}$. Substituting (12):

$$dh_{1G} = \alpha_G(\mathbf{G}\mathbf{C}_b - \mathbf{Q}_b)\mathbf{C}_b^T = \alpha_G(\mathbf{Q}_b\mathbf{C}_b^+\mathbf{C}_b\mathbf{C}_b^T - \mathbf{Q}_b\mathbf{C}_b^T)$$

which is $\mathbf{0}$, as $\mathbf{C}_b^+\mathbf{C}_b\mathbf{C}_b^T = \mathbf{C}_b^T(\mathbf{C}_b\mathbf{C}_b^T)^{-1}(\mathbf{C}_b\mathbf{C}_b^T) = \mathbf{C}_b^T$,

$$dh_{1H} = \alpha_H(\mathbf{H}\mathbf{C}_b - \mathbf{Q}_b)\mathbf{C}_b^T = \alpha_H(\mathbf{H}\mathbf{C}_b\mathbf{C}_b^T - \mathbf{Q}_b\mathbf{C}_b^T).$$

Therefore: $\dot{h}_1 = (\alpha_H/c_s)tr(\mathbf{H}\mathbf{C}_b\mathbf{C}_b^T - \mathbf{Q}_b\mathbf{C}_b^T) = 0$, using (22). One can also find in an analogous manner $\dot{h}_2 = 0$. Therefore $\dot{\mathbf{H}} = \mathbf{0}$, i.e., s_h and $\mathbf{R}_h$ do not change. Finally, we see that the centroid $\mathbf{q}_0$ is not changed by the controller:

$$\dot{\mathbf{q}}_0 = \frac{1}{N}\dot{\mathbf{Q}} \cdot \mathbf{1}_N = \frac{1}{N}(\alpha_H(\mathbf{H}\mathbf{C}_b - \mathbf{Q}_b) + \alpha_G(\mathbf{G}\mathbf{C}_b - \mathbf{Q}_b))\mathbf{1}_N.$$

As by definition $\mathbf{C}_b \cdot \mathbf{1}_N = \mathbf{0}$ and $\mathbf{Q}_b \cdot \mathbf{1}_N = \mathbf{0}$, $\dot{\mathbf{q}}_0 = \mathbf{0}$.

Proof of Lemma 2: We first develop the two sides of the equality to be proven as:

$$\begin{aligned}&||\mathbf{A}||_F^2 + s_w^2||\mathbf{B}||_F^2 - 2s_w tr(\mathbf{A}^T\mathbf{R}_w\mathbf{B}) = \\ &||\mathbf{A}||_F^2 + ||\mathbf{M}\mathbf{B}||_F^2 - 2tr(\mathbf{A}^T\mathbf{M}\mathbf{B}) + \\ &||\mathbf{M}\mathbf{B}||_F^2 + s_w^2||\mathbf{B}||_F^2 - 2s_w tr((\mathbf{M}\mathbf{B})^T\mathbf{R}_w\mathbf{B}). \qquad (27)\end{aligned}$$

We show two identities next. The first one is:

$$\begin{aligned}&||\mathbf{M}\mathbf{B}||_F^2 = tr((\mathbf{M}\mathbf{B})^T\mathbf{M}\mathbf{B}) = tr(\mathbf{B}^T\mathbf{B}^{+T}\mathbf{A}^T\mathbf{M}\mathbf{B}) = \\ &tr(\mathbf{B}\mathbf{B}^T\mathbf{B}^{+T}\mathbf{A}^T\mathbf{M}) = tr(\mathbf{B}\mathbf{A}^T\mathbf{M}) = tr(\mathbf{A}^T\mathbf{M}\mathbf{B}), \quad (28)\end{aligned}$$

where we used the cyclic property of the trace of a product and $\mathbf{B}\mathbf{B}^T\mathbf{B}^{+T} = \mathbf{B}\mathbf{B}^T(\mathbf{B}\mathbf{B}^T)^{-T}\mathbf{B} = \mathbf{B}$ due to the symmetry of $\mathbf{B}\mathbf{B}^T$. The second identity is obtained analogously:

$$\begin{aligned}&tr((\mathbf{M}\mathbf{B})^T\mathbf{R}_w\mathbf{B}) = tr(\mathbf{B}^T\mathbf{B}^{+T}\mathbf{A}^T\mathbf{R}_w\mathbf{B}) = \\ &tr(\mathbf{B}\mathbf{B}^T\mathbf{B}^{+T}\mathbf{A}^T\mathbf{R}_w) = tr(\mathbf{B}\mathbf{A}^T\mathbf{R}_w) = tr(\mathbf{A}^T\mathbf{R}_w\mathbf{B}). \quad (29)\end{aligned}$$

Substitution of (28) and (29) in (27) shows (27) is true.

Proof of Proposition 2: Prop. 1 showed that $\mathbf{U}_\gamma$ is decoupled as it does not change $\mathbf{q}_0$, θ_h, or s_h. Next we show $\mathbf{U}_0$, $\mathbf{U}_\theta$, $\mathbf{U}_s$ are also decoupled by computing the changes they produce in the relevant variables (γ, $\mathbf{q}_0$, θ_h, s_h).

- γ **decoupling.** We will show $\mathbf{U}_0$, $\mathbf{U}_\theta$, $\mathbf{U}_s$ do not change γ. The change in γ under a given control $\mathbf{U}_p$ is $tr((\mathbf{Q}_b - \mathbf{H}\mathbf{C}_b)^T\mathbf{U}_p)$. We evaluate this expression next replacing $\mathbf{U}_p$ by the different control terms.

$\underline{\mathbf{U}_0}$. We have $-\alpha_0 tr(\mathbf{K}_b(\mathbf{Q} - \mathbf{H}\mathbf{C})^T\mathbf{e}_0\mathbf{1}_N^T)$. Applying the cyclic property, this is equal to $-\alpha_0 tr(\mathbf{1}_N^T\mathbf{K}_b(\mathbf{Q} - \mathbf{H}\mathbf{C})^T\mathbf{e}_0)$, and by definition $\mathbf{1}_N^T\mathbf{K}_b = 0$.

$\underline{\mathbf{U}_\theta}$. Replacing $\mathbf{U}_p$ by $\mathbf{U}_\theta$ and using (25) we can see that $-\alpha_\theta e_\theta(tr(\mathbf{Q}_b^T\mathbf{S}\mathbf{H}\mathbf{C}_b) - tr((\mathbf{H}\mathbf{C}_b)^T\mathbf{S}\mathbf{H}\mathbf{C}_b)) = 0$.

$\underline{\mathbf{U}_s}$. Replacing $\mathbf{U}_p$ by $\mathbf{U}_s$ and using (24) we can see that $-\alpha_s e_s(1/s_h)(tr(\mathbf{Q}_b^T\mathbf{H}\mathbf{C}_b) - tr((\mathbf{H}\mathbf{C}_b)^T\mathbf{H}\mathbf{C}_b)) = 0$.

- $\mathbf{q}_0$ **decoupling**. We will show $\mathbf{U}_\theta$ and $\mathbf{U}_s$ do not change $\mathbf{q}_0$. For this we study the change in $\mathbf{q}_0$ under a given control $\mathbf{U}_p$. This is $\dot{\mathbf{q}}_0 = (1/N)\mathbf{U}_p \cdot \mathbf{1}_N$. For both $\mathbf{U}_\theta$ and $\mathbf{U}_s$, this results in a product $\mathbf{C}_b \cdot \mathbf{1}_N$, which is zero.

We will use in the next steps the changes in h_1 and h_2 under a control $\mathbf{U}_p$, which have the expressions:

$$\dot{h}_1 = (1/c_s)tr(\mathbf{U}_p\mathbf{K}_b\mathbf{C}_b^T), \qquad (30)$$

$$\dot{h}_2 = (1/c_s)tr(\mathbf{U}_p\mathbf{K}_b(\mathbf{S}\mathbf{C}_b)^T). \qquad (31)$$

- θ_h **decoupling.** We will show $\mathbf{U}_0$ and $\mathbf{U}_s$ do not change θ_h. The dynamics of θ_h is $\dot{\theta}_h = (h_1\dot{h}_2 - \dot{h}_1h_2)/s_h^2$.

$\underline{\mathbf{U}_0}$. We have $\dot{h}_1 = -(\alpha_0/c_s)tr(\mathbf{e}_0\mathbf{1}_N^T\mathbf{K}_b\mathbf{C}_b^T)$, where $\mathbf{1}_N^T\mathbf{K}_b = 0$. For this same reason, $\dot{h}_2 = 0$, and hence $\dot{\theta}_h = 0$.

$\underline{\mathbf{U}_s}$. We have $\dot{h}_1 = -\frac{\alpha_s e_s}{s_h c_s}tr(\mathbf{H}\mathbf{C}_b\mathbf{C}_b^T) = -\frac{\alpha_s e_s h_1}{s_h}$ (using (22)) and $\dot{h}_2 = -\frac{\alpha_s e_s}{s_h c_s}tr(\mathbf{H}\mathbf{C}_b(\mathbf{S}\mathbf{C}_b)^T) = -\frac{\alpha_s e_s h_2}{s_h}$ (using (23)). We conclude that $\dot{\theta}_h = 0$.

- s_h **decoupling.** We will show $\mathbf{U}_0$ and $\mathbf{U}_\theta$ do not change s_h. The dynamics of s_h is $\dot{s}_h = (\dot{h}_1h_1 + \dot{h}_2h_2)/s_h$.

$\underline{\mathbf{U}_0}$. Using the same argument as for the θ_h decoupling above, this control term does not change s_h.

$\underline{\mathbf{U}_\theta}$. We have $\dot{h}_1 = -(\alpha_\theta e_\theta/c_s)tr(\mathbf{S}\mathbf{H}\mathbf{C}_b\mathbf{C}_b^T)$, and using $\mathbf{S}^T = -\mathbf{S}$, the cyclic property, and (23) gives $\dot{h}_1 = \alpha_\theta e_\theta h_2$. On the other hand, $\dot{h}_2 = -(\alpha_\theta e_\theta/c_s)tr(\mathbf{S}\mathbf{H}\mathbf{C}_b(\mathbf{S}\mathbf{C}_b)^T)$. Using $\mathbf{S}^T = -\mathbf{S}$, $\mathbf{S}^2 = -\mathbf{I}_2$, the cyclic property, and (22) gives $\dot{h}_2 = -\alpha_\theta e_\theta h_1$. Hence $\dot{s}_h = 0$.

Proof of Theorem 1: From Prop. 2, the evolution of each error variable is only influenced by a single control term.

Convergence of γ. The evolution of γ is as follows:

$$\begin{aligned}\dot{\gamma} &= tr((\nabla_{\mathbf{Q}}\gamma)^T\mathbf{U}_\gamma) = -tr(\mathbf{U}_H^T(\alpha_H\mathbf{U}_H + \alpha_G\mathbf{U}_G)) \\ &= -2\alpha_H\gamma - \alpha_G tr(\mathbf{U}_H^T\mathbf{U}_G). \qquad (32)\end{aligned}$$

Here, we know directly from Lemma 2 –with $\mathbf{A} = \mathbf{Q}_b$, $\mathbf{B} = \mathbf{C}_b$, $\mathbf{H}_w = \mathbf{H}$–, that: $||\mathbf{U}_H||_F^2 = ||\mathbf{U}_G||_F^2 + ||\mathbf{U}_H - \mathbf{U}_G||_F^2$. Since by definition $||\mathbf{U}_H||_F^2 = ||\mathbf{U}_G||_F^2 + ||\mathbf{U}_H - \mathbf{U}_G||_F^2 - 2tr(\mathbf{U}_G^T(\mathbf{U}_H - \mathbf{U}_G))$, it must therefore hold that $tr(\mathbf{U}_G^T(\mathbf{U}_H - \mathbf{U}_G)) = 0$. Hence, $tr(\mathbf{U}_G^T\mathbf{U}_H) = tr(\mathbf{U}_H^T\mathbf{U}_G) = ||\mathbf{U}_G||_F^2 \geq 0$. From (32), this implies that $\dot{\gamma} \leq 0$, and that $\dot{\gamma} = 0$ can only happen when $\gamma = 0$ and $\mathbf{U}_G = 0$. But $\gamma = 0$ implies the robots are in the desired shape (Lemma 1), which means $\mathbf{U}_G = 0$. Hence, we can state the simple condition that $\dot{\gamma} = 0$ if and only if $\gamma = 0$.

$s_h(t)$ will be upper bounded for all t, since it can be fully controlled by $\mathbf{U}_s$. γ is also upper bounded –by its initial value– . As γ can be expressed as $\gamma = (1/2)||(\mathbf{Q} - s_h\mathbf{R}_h\mathbf{C})\mathbf{K}_b||_F^2$, $||\mathbf{Q}||_F$ must be upper bounded too: $||\mathbf{Q}||_F \leq \tilde{q}$ for a $\tilde{q} \in \mathbb{R}$. Defining $\mathbf{q} = vec(\mathbf{Q}) \in \mathbb{R}^{2N}$, this means that the compact set $\Omega = \{\mathbf{q} \text{ s.t. } ||\mathbf{q}|| \leq \tilde{q}\}$ is invariant. Then, as $\dot{\gamma}(\mathbf{q}) \leq 0$, LaSalle's invariance principle implies that $\mathbf{q}$ converges to the largest invariant set in Ω in which $\dot{\gamma} = 0$. As noted above, $\dot{\gamma} = 0$ implies $\gamma = 0$; hence, $\gamma \to 0$ and the system converges to the desired shape (Lemma 1). And from (32), if $\alpha_G = 0$, γ decreases exponentially, as $\dot{\gamma} = -2\alpha_H\gamma$.

Exponential decrease of $\mathbf{e}_0$**,** e_θ**,** e_s.

$\underline{\mathbf{e}_0}$: Recall that $\mathbf{q}_0 = \frac{1}{N}\mathbf{Q}\mathbf{1}_N$. So when applying $\mathbf{U}_0$ we get $\dot{\mathbf{e}}_0 = \dot{\mathbf{q}}_0 = \frac{1}{N}\mathbf{U}_0\mathbf{1}_N = -\frac{\alpha_0}{N}\mathbf{e}_0\mathbf{1}_N^T\mathbf{1}_N = -\alpha_0\mathbf{e}_0$.

$\underline{e_\theta}$: We already found (proof of Prop. 2) that under the action of $\mathbf{U}_\theta$, $\dot{h}_1 = \alpha_\theta e_\theta h_2$ and $\dot{h}_2 = -\alpha_\theta e_\theta h_1$. Therefore $\dot{\theta}_h = (h_1\dot{h}_2 - \dot{h}_1h_2)/s_h^2 = -\alpha_\theta e_\theta$. Hence, $\dot{e}_\theta = \dot{\theta}_h = -\alpha_\theta e_\theta$.

$\underline{e_s}$: We already found (proof of Prop. 2) that under the action of $\mathbf{U}_s$, $\dot{h}_1 = -\frac{\alpha_s e_s h_1}{s_h}$ and $\dot{h}_2 = -\frac{\alpha_s e_s h_2}{s_h}$. Therefore, $\dot{s}_h = (\dot{h}_1h_1 + \dot{h}_2h_2)/s_h = -\alpha_s e_s$. Hence, $\dot{e}_s = \dot{s}_h = -\alpha_s e_s$.

Therefore, all errors γ, $\mathbf{e}_0$, e_θ, e_s, are driven to zero as $t \to \infty$. $\gamma = 0$ implies that $\mathbf{Q}_b = \mathbf{H}\mathbf{C}_b$ (5). $e_\theta = 0$ and $e_s = 0$ imply that $\mathbf{H} = s_d\mathbf{R}(\theta_d)$. And $\mathbf{e}_0 = 0$ implies $\mathbf{Q}_0 = \mathbf{Q}_{0d}$. Since $\mathbf{Q} = \mathbf{Q}_b + \mathbf{Q}_0$ (1), $\mathbf{Q}$ converges to the target configuration $\mathbf{Q}_T = s_d\mathbf{R}(\theta_d)\mathbf{C}_b + \mathbf{Q}_{0d}$.

References

[1] A. Farinelli, L. Iocchi, and D. Nardi, "Multirobot systems: a classification focused on coordination," *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 34, no. 5, pp. 2015–2028, 2004.

[2] J. Sanchez, J.-A. Corrales, B.-C. Bouzgarrou, and Y. Mezouar, "Robotic manipulation and sensing of deformable objects in domestic and industrial applications: a survey," *The International Journal of Robotics Research*, vol. 37, no. 7, pp. 688–716, 2018.

[3] R. Herguedas, G. López-Nicolás, R. Aragüés, and C. Sagüés, "Survey on multi-robot manipulation of deformable objects," in *IEEE Int. Conf. on Emerging Technol. and Factory Autom. (ETFA)*, 2019, pp. 977–984.

[4] K.-K. Oh, M.-C. Park, and H.-S. Ahn, "A survey of multi-agent formation control," *Automatica*, vol. 53, pp. 424–440, 2015.

[5] J. Zhu, B. Navarro, P. Fraisse, A. Crosnier, and A. Cherubini, "Dual-arm robotic manipulation of flexible cables," in *IEEE/RSJ Intern. Conf. on Intelligent Robots and Systems (IROS)*, 2018, pp. 479–484.

[6] M. Müller, B. Heidelberger, M. Teschner, and M. Gross, "Meshless deformations based on shape matching," *ACM Trans. Graph.*, vol. 24, no. 3, pp. 471–478, 2005.

[7] J. Alonso-Mora, R. Knepper, R. Siegwart, and D. Rus, "Local motion planning for collaborative multi-robot manipulation of deformable objects," in *IEEE Int. Conf. Rob. and Aut. (ICRA)*, 2015, pp. 5495–5502.

[8] D. Kruse, R. J. Radke, and J. T. Wen, "Human-robot collaborative handling of highly deformable materials," in *American Control Conference (ACC)*, 2017, pp. 1511–1516.

[9] I. G. Ramirez-Alpizar, K. Harada, and E. Yoshida, "Motion planning for dual-arm assembly of ring-shaped elastic objects," in *IEEE-RAS Int. Conf. on Humanoid Robots (HUMANOIDS)*, 2014, pp. 594–600.

[10] J. Das and N. Sarkar, "Autonomous shape control of a deformable object by multiple manipulators," *Journal of Intelligent & Robotic Systems*, vol. 62, no. 1, pp. 3–27, 2011.

[11] F. Ficuciello, A. Migliozzi, E. Coevoet, A. Petit, and C. Duriez, "FEM-based deformation control for dexterous manipulation of 3D soft objects," in *IEEE/RSJ International Conference on Intelligent Robots and Sytems (IROS)*, 2018, pp. 4007–4013.

[12] 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.

[13] R. Ritz and R. D'Andrea, "Carrying a flexible payload with multiple flying vehicles," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2013, pp. 3465–3471.

[14] R. Cotsakis, D. St-Onge, and G. Beltrame, "Decentralized collaborative transport of fabrics using micro-UAVs," in *IEEE Int. Conf. on Robotics and Automation (ICRA)*, 2019, pp. 7734–7740.

[15] M. Mukadam, A. Borum, and T. Bretl, "Quasi-static manipulation of a planar elastic rod using multiple robotic grippers," in *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, 2014, pp. 55–60.

[16] D. Navarro-Alarcón, Y. Liu, J. G. Romero, and P. Li, "Model-free visually servoed deformation control of elastic objects by robot manipulators," *IEEE Trans. Rob.*, vol. 29, no. 6, pp. 1457–1468, 2013.

[17] Z. Hu, P. Sun, and J. Pan, "Three-dimensional deformable object manipulation using fast online Gaussian process regression," *IEEE Robotics and Automation Letters*, vol. 3, no. 2, pp. 979–986, 2018.

[18] 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, 2015.

[19] J. T. Shepard and C. A. Kitts, "A multirobot control architecture for collaborative missions comprised of tightly coupled, interconnected tasks," *IEEE Systems Journal*, vol. 12, no. 2, pp. 1435–1446, 2018.

[20] Z. Cao, N. Gu, J. Jiao, S. Nahavandi, C. Zhou, and M. Tan, "A novel geometric transportation approach for multiple mobile manipulators in unknown environments," *IEEE Systems Journal*, vol. 12, no. 2, pp. 1447–1455, 2018.

[21] L. Zhou, V. Tzoumas, G. J. Pappas, and P. Tokekar, "Resilient active target tracking with multiple robots," *IEEE Robotics and Automation Letters*, vol. 4, no. 1, pp. 129–136, 2019.

[22] D. Lee and M. W. Spong, "Stable flocking of multiple inertial agents on balanced graphs," *IEEE Transactions on Automatic Control*, vol. 52, no. 8, pp. 1469–1475, 2007.

[23] 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, 2008.

[24] M. Aranda, Y. Mezouar, G. López-Nicolás, and C. Sagüés, "Enclosing a moving target with an optimally rotated and scaled multiagent pattern," *International Journal of Control*, vol. 94, no. 3, pp. 601–611, 2021.

[25] K. Fathian, S. Safaoui, T. H. Summers, and N. R. Gans, "Robust 3D distributed formation control with collision avoidance and application to multirotor aerial vehicles," in *IEEE Int. Conf. on Robotics and Automation (ICRA)*, 2019, pp. 9209–9215.

[26] 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, 2009.

[27] H. Garcia de Marina, B. Jayawardhana, and M. Cao, "Distributed rotational and translational maneuvering of rigid formations and their applications," *IEEE Trans. Rob.*, vol. 32, no. 3, pp. 684–697, 2016.

[28] M. D. Queiroz, X. Cai, and M. Feemster, *Formation Control of Multi-Agent Systems: A Graph Rigidity Approach*. Hoboken, NJ, USA: Wiley, 2019.

[29] 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, 2016.

[30] X. Luo, X. Li, X. Li, J. Yan, and X. Guan, "Globally stable formation control of nonholonomic multiagent systems with bearing-only measurement," *IEEE Systems Journal*, vol. 14, no. 2, pp. 2901–2912, 2020.

[31] Z. Wang and M. Schwager, "Force-Amplifying N-robot Transport System (force-ANTS) for cooperative planar manipulation without communication," *The Int. Journ. of Robotics Research*, vol. 35, no. 13, pp. 1564–1586, 2016.

[32] H. Bai and J. T. Wen, "Cooperative load transport: A formation-control perspective," *IEEE Trans. on Rob.*, vol. 26, no. 4, pp. 742–750, 2010.

[33] D. Sieber and S. Hirche, "Human-guided multirobot cooperative manipulation," *IEEE Transactions on Control Systems Technology*, vol. 27, no. 4, pp. 1492–1509, 2019.

[34] X. Dong, B. Yu, Z. Shi, and Y. Zhong, "Time-varying formation control for unmanned aerial vehicles: Theories and applications," *IEEE Trans. Contr. Syst. Technol.*, vol. 23, no. 1, pp. 340–348, 2015.

[35] 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, 2016.

[36] S. Zhao, "Affine formation maneuver control of multiagent systems," *IEEE Trans. Autom. Control*, vol. 63, no. 12, pp. 4140–4155, 2018.

[37] T. George Thuruthel, Y. Ansari, E. Falotico, and C. Laschi, "Control strategies for soft robotic manipulators: a survey," *Soft Robotics*, vol. 5, no. 2, pp. 149–163, 2018.

[38] A. Chen, R. Yin, L. Cao, C. Yuan, H. K. Ding, and W. J. Zhang, "Soft robotics: definition and research issues," in *24th International Conference on Mechatronics and Machine Vision in Practice (M2VIP)*, 2017, pp. 366–370.

[39] R. Yin, K. Wang, S. Du, L. Chen, J. Nie, and W. Zhang, "Design of genipin-crosslinked microgels from concanavalin a and glucosyloxyethyl acrylated chitosan for glucose-responsive insulin delivery," *Carbohydrate Polymers*, vol. 103, pp. 369–376, 2014.

[40] T. Remani, E. A. Jasmin, and T. P. I. Ahamed, "Residential load scheduling with renewable generation in the smart grid: a reinforcement learning approach," *IEEE Systems Journal*, vol. 13, no. 3, pp. 3283–3294, 2019.

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

[42] W. J. Zhang and C. Van Luttervelt, "Towards a resilient manufacturing system," *CIRP Annals*, vol. 60, no. 1, pp. 469–472, 2011.

[43] T. Zhang, W. Zhang, and M. M. Gupta, "Resilient robots: concept, review, and future directions," *Robotics*, vol. 6, no. 4, 2017.

[44] F. Wang, Z. Qian, Z. Yan, C. Yuan, and W. Zhang, "A novel resilient robot: kinematic analysis and experimentation," *IEEE Access*, vol. 8, pp. 2885–2892, 2020.

[45] 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. Cont.*, vol. 63, no. 5, pp. 1509–1516, 2018.

[46] X. Hu, L. Cao, Y. Luo, A. Chen, E. Zhang, and W. J. Zhang, "A novel methodology for comprehensive modeling of the kinetic behavior of steerable catheters," *IEEE/ASME Transactions on Mechatronics*, vol. 24, no. 4, pp. 1785–1797, 2019.