Citation: I. Cuiral-Zueco and G. López-Nicolás, "Multiscale Procrustes-Based 3-D Shape Control," in IEEE/ASME Transactions on Mechatronics, doi: 10.1109/TMECH.2023.3325934.

Multiscale Procrustes-based 3-D shape control

Ignacio Cuiral-Zueco and Gonzalo López-Nicolás

***Abstract*—The shape control problem involves deforming a non-rigid object into a desired shape. In this paper, we present a new 3D shape control method that can handle large deformations of texture-less objects. In particular, we use functional maps to generate surface points' correspondences between the current and the desired shape that serve as input for our Procrustes-based multi-scale control strategy. The main contribution is a multi-scale analysis with a relaxed rigidity assumption that defines scales based on geodesic distances to the grippers. Our method considers that objects may deform at different scales depending on the gripper actions applied to them. For this reason, our multi-scale analysis computes shape control actions defined according to gripper influence on particularly convenient scales. We present stability analysis and discuss several simulations and experiments.**

Keywords*—Shape control, deformable object manipulation, multi-robot manipulation, shape servoing.

I. Introduction

AUTOMATING object deformation processes has multiple applications such as manipulation-related tasks performed in industrial processes, surgical procedures or home-robotics. The wide variety of tasks and object types hinders the process of classifying and contextualising control strategies. This paper focuses on the shape control task, which is defined in surveys of the deformable object manipulation literature (e.g., [1]–[3]). Within the deformable object manipulation context, these surveys propose different criteria for the classification of objects, control strategies or perception methods. In [2], a deformable object classification is proposed. Based on physical and shape criteria, [2] classifies objects into cloth-like, linear, planar and solid objects. Within the manipulation tasks, they include the shape control problem for planar objects and sub-divide it into single point and multiple points shape control. Survey [3] includes a classification of objects according to their spatial dimensions and categorises them into 1D, 2D and 3D objects. Regarding the deformable object perception, [1] classifies the perception of deformable objects into: force-based, vision-based, and both vision and force-based perception. Focusing more on vision-based sensing, [3] includes control-related concepts such as state estimation, model, state-template or parameter identification in its perception section.

A. Related work

Some shape control methods rely on the use of Jacobian matrices that model the object deformation dynamics in a local manner. This is the case of [4] in which the stretch limits of the objects and gripper collisions are considered. In [5], the Jacobian matrix is estimated from visual-features and used to control feedback points of elastic objects. In [6] and [7], the authors exploit the As-Rigid-As-Possible deformation model (ARAP [8]) in order to achieve shape control. Both approaches are validated through several experiments. Another Jacobian computation approach is presented in [9], where the Jacobian matrix parameters are estimated with the use of truncated Fourier series of the 2D object's contour (i.e. the object's 1D closed contour embedded in 2D). The method is validated in experiments that involve an active and a passive gripper. Using Fourier series as well, dual-arm flexible cable manipulation experiments are presented in [10]. Cable manipulation is also tackled in [11], [12] and [13]. In the former, the main-connectivity of the cable's free configuration space is captured by a simplified pre-computed descriptor that allows to numerically solve the cable's ODE at a reduced cost. The method in [12] presents a B-spline based model that relies on 3D tracking wires with the use of a particle filter. The system proposed in [13] models collisions, contacts and frictions between cables and a plane in the workspace.

Making use of an adaptive deformation model, experiments involving materials such as foam, meat or plastic are carried out satisfactorily in [14]. In [15], a Principal Component Analysis (PCA) is applied to the 2D contour of the object's silhouette. In [16], real experiments involving large isometric deformations on planar objects are carried out. This method uses monocular perception and a Shape-from-Template (SfT) based algorithm. Image contour moments are used to define a sliding control strategy to control the shape of objects that range from soft to rigid (articulated) [17]. Their method is validated with a stability proof and experiments with a dual-arm robot setup.

In [18], a 3D deformable object servo control based on a Gaussian Process Regression (GPR) online learned model is presented. The method in [19] focuses on deformable object transport and introduces a consensus-based deformation model for the manipulation of broad flexible objects with the use of a large number of manipulators. Two heuristic methods and a neural network for shaping non-prehensile materials with plastic deformation characteristics (not elastic) are introduced and compared in [20]. They acquire desired shapes by pushing kinetic sand with adapted robotic arms. For more state of the art publications we refer the reader to the special issue [21] which collects a variety of publications focusing on the manipulation of deformable objects.

B. Proposed method and contributions

In the deformable object manipulation literature, some methods focus on specific tasks (e.g., object cutting [22]) or specific

I. Cuiral-Zueco and G. López-Nicolás are with Instituto de Investigación en Ingeniería de Aragón, Universidad de Zaragoza, Spain. ignaciocuiral@unizar.es, gonlopez@unizar.es. This work was supported 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.

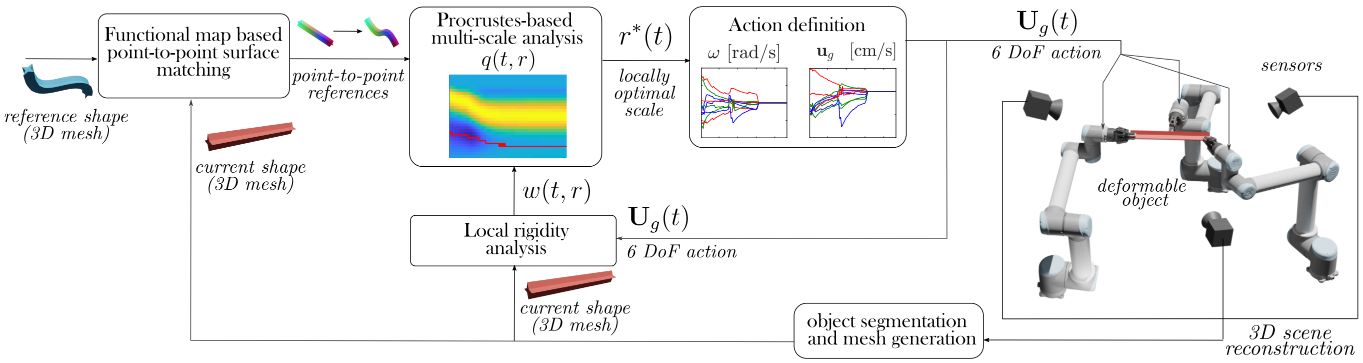

Fig. 1: Control scheme of the shape control method. A 3D object mesh is extracted from the scene's 3D reconstruction. Using functional maps, the current shape mesh is matched to the reference (or target) shape mesh. A multi-scale analysis is then performed making use of a local rigidity measurement of how rigidly the object behaves at each scale. The multi-scale analysis provides an optimal scale value that serves as input for our control law.

object types (e.g., flexible PCB manipulation in [23]). In this paper we propose a method for shape control that tackles the manipulation of texture-less objects that may undergo large 3D deformations (i.e. object's curvature, length, area or volume can globally vary in 3D space). Note that we focus on shape control, meaning that object transport (e.g., considering error with a rigid component such as an arbitrary translation and rotation of the object) is outside our scope. The goal is to define a control system that allows to manipulate the shape of a deformable object towards a desired target shape with the use of robots and vision sensors (Fig. 1).

Objects are assumed to be large-strain (i.e. to have low Young's modulus) and we assume there is no information about their specific physical properties (density, stiffness, etc.). We focus on objects that present certain rigidity such that gravity or inertia do not dominate their behaviour (e.g. clothes are not considered). Our assumptions include proper object perception (no significant occlusions) and knowledge of the initial gripper configuration. While better results might be achieved with appropriate gripper configuration, our method does not rely on the assumption that the grippers are ideally positioned on the object. The effects of gravity and inertia are disregarded given the assumption that deformations are gradual and slow. As the objects may be texture-less, our method does not rely on visual descriptors.

Regarding the control strategy (see Fig. 1), in section II we propose a multi-scale analysis to determine the scale on which our novel control strategy should be focused at each time instant during the deformation process (scale defined as the topological distance at which surface points lie from a gripper). We do so under what we defined as the local rigidity assumption. We validate the assumption by means of a multi-scale Procrustes analysis, i.e. we measure the extent to which the object can be considered to be moving rigidly at each scale. Using this information, we define a 6 DoF control action for each one of the grippers that are manipulating the object. We provide a stability analysis (section III) and several simulation and experiments (section IV) that validate our proposed system.

In the following, we will discuss and analyse the main contributions and contextualise them in the literature.

- We define a novel shape control framework for 3D deformations of both planar and volumetric objects. Our method is the first 3D shape control system that, with the use of functional maps, performs a holistic shape analysis. That is, thanks to the novel application of [24] in a real shape control setup, we define our 3D control strategy and shape error by analysing all of the object's perceived geometry (regardless of its visual texture) and consider deformations at different scales. Most existing methods tackle 1D (linear) objects (e.g. [12], [13], [25]–[27]) or confine their analysis to 2D contours (e.g., [9], [15], [17], [28]). Few methods address 3D deformations of planar and/or volumetric objects, e.g. [14], [18], [29]. Rather than addressing holistic shape control, these methods present task-oriented approaches. For example, they base their shape analysis and errors on a few discrete features such as a segment's curvature, the object's centroid or the position of a reduced number of feature points.
- We prove local asymptotic stability of our control system and further validate the method with experiments. Related works also study local stability [14], [18]; other works such as [29] do not provide stability analysis nor guarantees system convergence.
- We validate our proposed model using real data from our experiments and compare it against other existing baselines. The comparisons show that our model is suitable for shape control given its proper balance between accuracy and computational time cost.
- Our proposed control method does not require a priori information or initial exploration of the object's behaviour. Although online-updated, methods such as [14] or [18] require an initial exploration of the object behaviour and/or a random initialisation before converging to what they refer to as a *good enough initial model*.

II. Multi-scale Procrustes shape control

In this section we present a shape control strategy that, for each gripper $g = 1, ..., G$, defines a 6 DoF gripper control action expressed as a rigid transform $\mathbf{U}_g \in SE(3)$ (composed by translation action in $\mathbf{u}_g \in \mathbb{R}^3$ and rotational action in Euler angles $\omega \in \mathbb{R}^3$).

A. Procrustes operator and shape error metric.

Functional maps [24], based on the spectral analysis of the mesh through the Laplace-Beltrami operator, allow us to obtain a point-to-point match between current shape and target shape mesh nodes. We denote the current shape mesh nodes positions by $\mathbf{x}_m \in \mathbb{R}^3, m = 1, ..., M$. These vectors are stacked in matrix $\mathbf{X} \in \mathbb{R}^{3\times M}$. Each current shape point $\mathbf{x}_m$ has an associated (matched) target point $\mathbf{y}_m \in \mathbb{R}^3$. These target points are column-wise stacked in $\mathbf{Y} \in \mathbb{R}^{3\times M}$.

For simplicity of notation, we define the Procrustes operator $(\mathbf{T}, d_{\mathcal{P}}) = \mathcal{P}(\mathbf{X}, \mathbf{Y})$. This operator encloses the orthogonal Procrustes problem as it takes the two column-to-column matched sets of point coordinates $\mathbf{X}, \mathbf{Y}$ and returns their Procrustes distance $d_{\mathcal{P}}$ and the rigid transform $\mathbf{T}(\mathbf{X}, \mathbf{Y}) \in SE(3)$ that minimises such distance:

$$d_{\mathcal{P}}(\mathbf{X}, \mathbf{Y}) = \min_{\mathbf{R}} \left\| \mathbf{R}(\mathbf{X} - \bar{\mathbf{X}}) - (\mathbf{Y} - \bar{\mathbf{Y}}) \right\|_F \qquad \text{s.t.}\, \mathbf{R} \in \mathrm{SO}(3). \tag{1}$$

Matrices $\bar{\mathbf{X}} \in \mathbb{R}^{3\times M}$ and $\bar{\mathbf{Y}} \in \mathbb{R}^{3\times M}$ stack the column-wise mean $\bar{\mathbf{x}}, \bar{\mathbf{y}}$ (i.e. the centroid) of matrices $\mathbf{X}, \mathbf{Y}$. Matrix $\mathbf{R} \in \mathrm{SO}(3)$ is the rotation component of $\mathbf{T}$ and $\mathbf{t} = \bar{\mathbf{y}} - \mathbf{R}\bar{\mathbf{x}}$ is the translation component.

We can apply the Procrustes operator $\mathcal{P}(\mathbf{X}(t), \mathbf{Y})$ to obtain shape error

$$e(t) = d_{\mathcal{P}}(\mathbf{X}(t), \mathbf{Y}), \tag{2}$$

which measures how similar shapes are ($e(t) = 0$ when two shapes are identical). The goal of our control strategy is to reduce the error metric $e(t)$. Before initiating our control strategy, we apply $\mathbf{T}^{-1}(t_0)$, obtained from $\mathcal{P}(\mathbf{X}(t_0), \mathbf{Y})$, to our target shape as to bring it closer to our current shape in the 3D embedding.

B. Local-rigidity behaviour (LRB) hypothesis

Consider we were only focusing on the surface points that lie within a topological distance r from a gripper g, i.e. a set of M_g object points $\mathbf{X}_g(t, r) \in \mathbb{R}^{3\times M_g}$ defined by points $\mathbf{x}_m(t, r) \in \Omega_g(t, r)$, where $\Omega_g(t, r)$ is the object's surface open domain defined by a geodesic ball of radius r centred at gripper g. Suppose the object's rigidity (unknown for us) allows for points $\mathbf{X}_g(t, r)$ to move on a rigid manner under small gripper transforms $\mathbf{H}_g$, i.e. $\mathbf{H}_g \approx \mathbf{I}_4$ (being $\mathbf{I}_4$ the 4×4 identity matrix), and the rest of points in $\mathbf{X}(t)$, i.e. $\mathbf{x}_m(t) \notin \Omega_g(t, r)$, remain unaffected by $\mathbf{H}_g$. We denote this rigid behaviour as local-rigidity behaviour (LRB for now on). In this scenario, one could benefit from a Procrustes analysis $\mathcal{P}(\mathbf{X}_g(t, r), \mathbf{Y}_g(r))$, where $\mathbf{Y}_g(r) \in \mathbf{Y}$ are the points matched to those of $\mathbf{X}_g(t, r)$. The transform $\mathbf{T}_g(t, r)$ from the Procrustes analysis can be used to define a incremental transform

$$\mathbf{H}_g(t, r) = \exp\left(\Delta t \log\left(\mathbf{T}_g(t, r)\right)\right). \tag{3}$$

where $\mathbf{H}_g(t, r)$ belongs to the geodesic path (in SE(3)) defined from $\mathbf{I}_4$ towards $\mathbf{T}_g(t, r)$. This path is parameterised by time step $\Delta t \in \mathbb{R}$, $\Delta t \in [0, 1]$ which, when taking low values, generates $\mathbf{H}_g(t, r)$ that meet the *small action* requirement for local rigid behaviour (LRB). Therefore, the rigid error reduction of the subset $\mathbf{X}_g(t, r)$ can be performed with actions $\mathbf{U}_g(t) = \mathbf{H}_g(t, r)$ and thus the global Procrustes residual of the whole set $\mathbf{X}(t)$ with respect to $\mathbf{Y}$, i.e. $e(t)$, can be reduced too. Note that, as $\Delta t \to 0$, (3) defines the state equation of $\mathbf{X}_g(t, r)$:

$$\mathbf{X}^h_g(t, r) = \mathcal{T}\{\prod_{t_0}^{t} e^{\log(\mathbf{T}_g(t,r))\mathrm{d}t}\}\mathbf{X}^h_g(t_0, r), \tag{4}$$

where h denotes homogeneous coordinates and $\mathcal{T}$ operates on the product integral generating a time-ordered product $\mathcal{T}\{\prod_{t_0}^{t} f(t)\} = f(t)f(t - \mathrm{d}t)\cdots f(t_0 + \mathrm{d}t)f(t_0)$. Equation (4) constitutes a solution for:

$$\frac{\mathrm{d}\mathbf{X}^h_g(t, r)}{\mathrm{d}t} = \log(\mathbf{T}_g(t, r))\mathbf{X}^h_g(t, r), \tag{5}$$

which defines the time derivative of points when they are only affected by gripper g.

C. Relaxed local-rigidity behaviour (LRB) analysis

In the diminishing rigidity concept introduced in [4], an exponential decay of the material's rigidity with respect to gripper positions is assumed. However, an object may present diverse and time-varying behaviours depending on its shape and/or deformation state (e.g., a discontinuous rigidity function as in a mechanism). Our method does not assume any particular rigidity decay function on the object. Rather, our proposed relaxed LRB assumption allows us to evaluate and quantify on which scale (or topological distance) gripper actions are more effective in reducing the shape error.

Local-rigidity behaviour (LRB) is certainly met by points grabbed by the grippers (assuming grasping stability). However, the rest of the object points will most likely undergo deformations and thus not present LRB. For this reason, we base our control strategy on a relaxed assumption of LRB, i.e., we make use of a multi-scale analysis that quantifies how close to the LRB our object behaviour is for each analysed scale. In order to perform a multi-scale analysis, we establish scale $r \in [r_0, R(t)]$ being r_0 the gripper's size and $R(t)$ the largest topological distance that can be found in the object. Our analysis quantifies the extent to which sets $\mathbf{X}_g(t, r)$ behave rigid-like under any action $\mathbf{U}_g(t)$. If actions $\mathbf{U}_g(t)$ affected $\mathbf{X}(t)$ at scale r with ideal LRB and assuming linear action superposition (given object's isotropy and homogeneity), we could estimate the resulting shape points $\hat{\mathbf{X}}(t, r)$ as:

$$\hat{\mathbf{x}}^h_m(t, r) = \frac{1}{G}\sum_{g}^{G}(\mathbf{U}_g(t))^{\delta_g(m,r)}\mathbf{x}^h_m(t, r) \tag{6}$$

where function $\delta_g(m,r)$ allows to disregard actions for points $\mathbf{x}_m(t,r) \notin \Omega_g(t,r)$:

$$\delta_g(m,r) := \begin{cases} 1 & \text{if } \mathbf{x}_m(t,r) \in \Omega_g(t,r) \\ 0 & \text{otherwise.} \end{cases} \tag{7}$$

Using the Procrustes analysis $\mathcal{P}(\mathbf{X}(t), \hat{\mathbf{X}}(t,r))$, we can obtain a measure $w(t,r)$ of how much the object presents LRB at each scale r (i.e. at each topological distance r from the grippers) when undergoing gripper actions $\mathbf{U}_g(t)$:

$$w(t,r) = 1/\exp(\beta\, d_{\mathcal{P}}(\mathbf{X}(t), \hat{\mathbf{X}}(t,r))). \tag{8}$$

Measure $w(t,r) \in (0,1]$, $w(t,r) = 1$ when the LRB is fully met (i.e. $d_{\mathcal{P}}(\mathbf{X}(t), \hat{\mathbf{X}}(t,r)) = 0$). Parameter $\beta > 0$ allows to modify the relaxation of the LRB assumption. Lower values of β imply a more relaxed rigidity assumption (e.g., if $\beta = 0.1$, almost every $w(t,r) \approx 1$ and thus we consider every set $\mathbf{X}_g(t,r), \forall r \in [r_0, R(t)]$ to move rigidly, even if they do not). We propose using $\beta \approx 1 \times 10^4$, which implies a conservative assumption of LRB.

D. Procrustes-based locally optimal scale estimation

Note that $w(t,r)$ also quantifies the effectiveness with which the rigid error of $\mathbf{X}_g(t,r)$ can be reduced by means of incremental transforms $\mathbf{H}_g(t,r)$ as defined in (3) (larger $w(t,r)$ implies more effectiveness). With this information, we seek to define gripper actions $\mathbf{U}_g(t)$ such that they better benefit the global error reduction $e(t)$. In order to define $\mathbf{U}_g(t)$, we propose analysing scenarios in $r \times r' \in \mathbb{R} \times \mathbb{R} : r, r' \in [r_0, R(t)]$. These scenarios constitute an estimation of the object evolution if it was affected by actions $\mathbf{U}_g(t) = \mathbf{H}_g(t,r)$ (defined at scale r) but presented ideal LRB at scale r'. Each estimation $\hat{\mathbf{X}}(t,r,r')$ is defined as:

$$\hat{\mathbf{x}}^h_m(t,r,r') = \frac{1}{G}\sum_g^G \mathcal{T}\{\prod_t^{t+\mathrm{d}t} (\mathbf{H}_g(t,r))^{\delta_g(m,r')}\}\mathbf{x}^h_m(t,r') \tag{9}$$

In order to perform our analysis, we define an error increment estimation surface

$$\hat{e}(t,r,r') = d_{\mathcal{P}}(\hat{\mathbf{X}}(t,r,r'), \mathbf{Y}) - e(t). \tag{10}$$

Surface $\hat{e}(t,r,r')$ constitutes a continuous surface that provides an insight on the effectiveness and risks in reducing error $e(t)$ by means of actions $\mathbf{U}_g(t) = \mathbf{H}_g(t,r)$ (generated under the ideal LRB assumption). We can incorporate our knowledge on how much the LRB is met at each scale r' (i.e. $w(t,r')$) and define an error-reduction effectiveness $q(t,r) \in \mathbb{R}$ which, for each estimated transform $\mathbf{H}_g(t,r)$, yields:

$$q(t,r) = \int_{r_0}^{R} \hat{e}(t,r,r')w(t,r')\mathrm{d}r'. \tag{11}$$

In (11), ideal error increment estimations $\hat{e}(t,r,r')$ are weighted by the error reduction effectiveness $w(t,r')$. In particular, $q(t,r)$ estimates the global error increment that each $\mathbf{H}_g(t,r)$ (3) can generate. The logical choice of r when defining gripper actions $\mathbf{U}_g(t) = \mathbf{H}_g(t,r)$ would be the one that ensures the largest error reduction, i.e. $\mathbf{U}_g(t) = \mathbf{H}_g(t, r^*(t))$, $r^*(t) = \arg\min_r(q(t,r))$. However, $w(t,r)$ (present in the definition of $q(t,r)$) is computed based on the effects of previous actions defined according to $\mathbf{U}_g(t) = \mathbf{H}_g(t, r^*(t))$. For $q(t,r)$ to be reliable, updates in $r^*(t)$ should take place in the locality of $r^*(t)$. For this reason we define $r^*(t)$ as:

$$r^*(t) = -\int_{t_0}^{t} \frac{\partial q(t, r^*(t))}{\partial r} + r^*(t_0), \tag{12}$$

which updates $r^*(t)$ in the direction of the $\partial/\partial r$ component of the gradient for a given time instant t and thus generates a locally optimal $r^*(t)$.

Note that (12) requires $q(t, r^*(t))$ to be continuous differentiable with respect to r. This implies $e(t), w(t,r')$ and $\hat{e}(t,r,r')$, and thus Procrustes transforms $\mathbf{T}_g$ and distances $d_{\mathcal{P}}$, should be continuous differentiable with respect to r. The residual $d_{\mathcal{P}}$ is continuous and differentiable as it constitutes a metric in shape space [30]. On the other hand, showing that the Procrustes optimisation result $\mathbf{T}_g$ is continuous and differentiable with respect to r requires more development that can be found in Appendix A.

We now provide some intuitions on impact of β in (8) on (12). When setting a low value for β (e.g., $\beta = 0.1$), we assume that our LRB hypothesis holds even for actions and scales that lie far from our current actions $\mathbf{U}_g(t)$ and scale $r^*(t)$. Consequently, surface $q(r,t)$ allows for a more unrestricted evolution of $r^*(t)$, which could potentially lead to undesired system behaviours such as slower performance or, in the case of extremely low values of β, to larger final error. On the other hand, a high β (e.g., $\beta = 1 \times 10^4$) leads to a conservative $q(r,t)$ surface that disregards estimates that lie far from the current actions and locally optimal scale. This conservative approach generates resistance to large changes in $r^*(t)$, confining its movement to regions where the LRB hypothesis has been validated through measurements across iterations. Very large β values will not compromise the system's effectiveness, but they will lead to slower convergence as $r^*(t)$ evolves more conservatively.

E. Control strategy

Our control strategy makes use of the Procrustes action defined at scale $r^*(t)$, i.e. $\mathbf{T}_g(t, r^*(t))$. We defined our control law as the new term in the time-ordered product of (4):

$$\mathbf{U}_g(t) = \exp\left(\log\left(\mathbf{T}_g(t, r^*(t))\right)\mathrm{d}t\right). \tag{13}$$

This results in the state equation of $\mathbf{X}(t)$ (for each individual $\mathbf{x}_m(t) \in \mathbf{X}(t)$)

$$\mathbf{x}^h_m(t) = \frac{1}{G}\sum_g^G \mathcal{T}\{\prod_{t_0}^{t} (\mathbf{U}_g(t))^{\delta_g(m, r^*(t))}\}\mathbf{x}^h_m(t_0, r^*(t_0)), \tag{14}$$

with $r^*(t)$ updated as in (12). Note that the update rule in (12) needs an initial value of $r^*(t_0)$ and $w(t_0, r')$. We propose $w(t_0, r') = 1 \forall r'$, which is equivalent to assuming equal LRB at all scales r'. Our initial estimation of r' takes the minimum at the initial time instant, that is $r^*(t_0) = \arg\min_r(q(t_0, r))$. Note that, for the update of $r^*(t)$, the partial derivative of $q(t, r^*(t))$ in (12) needs to be evaluated only at $r^*(t)$. This avoids the need to compute $q(t, r)$ in (11) for all r except for the neighbourhood of $r^*(t)$, contributing to the cost-effectiveness of our method. Furthermore, the rest of the method relies on matrix operations and SVD decomposition, further enhancing its low-cost nature, as it will be illustrated in the experiments section.

III. Stability analysis

In this section we prove that the contribution of each gripper to the error derivative $\dot{e}(t)$ presents local asymptotic stability. First, some preliminary concepts are presented. Then, we introduce several lemmas that lead to theorem III.4, which concludes local asymptotic stability of the system.

A. Preface

We use Tr to refer to the matrix's trace. For clarity, the dependence on r^* and t has been omitted when it could be easily inferred. Recall super-index h denotes homogeneous coordinates. The transpose of the logarithm of a rigid transform is (see Appendix A in [31]):

$$\log(\mathbf{T}_g^\intercal) = \log\left(\begin{array}{c|c} \mathbf{R}_g & \mathbf{t}_g \\ \mathbf{0}_{1\times 3} & 1 \end{array}\right) = \left(\begin{array}{c|c} \log(\mathbf{R}_g^\intercal) & \\ (\mathbf{t}')^\intercal & \mathbf{0}_{4\times 1} \end{array}\right), \tag{15}$$

where the rotation term $\log(\mathbf{R}_g^\intercal)$ can be obtained from:

$$\log(\mathbf{R}_g) = \frac{\theta}{2\sin(\theta)}(\mathbf{R}_g - \mathbf{R}_g^\intercal), \tag{16}$$

being $\theta \in (-\pi, \pi)$ the angle of rotation induced by $\mathbf{R}_g$. Translation term $\mathbf{t}' \in \mathbb{R}^3$ in (15) is $\mathbf{t}' = \mathbf{V}_g\mathbf{t}_g$ (recall $\mathbf{t}_g = \bar{\mathbf{y}}_g - \mathbf{R}_g\bar{\mathbf{x}}_g$), being $\mathbf{V}_g \in \mathbb{R}^{3\times 3}$:

$$\mathbf{V}_g = \mathbf{I}_3 - \frac{1}{2}\log(\mathbf{R}_g) + \frac{1}{\theta^2}\left(1 - \frac{\theta(1+\cos(\theta))}{2\,\sin(\theta)}\right)\log(\mathbf{R}_g)^2. \tag{17}$$

B. Lyapunov stability

Note that the Procrustes residual is invariant to changes in the frame of reference as transformations $\mathbf{T}_g$ can be expressed in any frame of reference. For convenience we defined G inertial frames of reference $\mathcal{F}_g(t)$ with axes always aligned with the global reference axes. Although their orientation does not change, frames $\mathcal{F}_g(t)$ present a changing position $\bar{\mathbf{x}}_g(t)$, being $\bar{\mathbf{x}}_g(t)$ the centroid of $\mathbf{X}_g(t, r^*(t))$. This is specially convenient as $\mathcal{F}_g(t)$ implies $\bar{\mathbf{x}}_g = \mathbf{0}, \forall\, t$ (for that particular reference). For now on, sub-index g implies point coordinates and transforms are defined from frame $\mathcal{F}_g(t)$.

Lemma III.1. $\mathrm{Tr}\left(\mathbf{X}_g^h(\mathbf{X}_g^h)^\intercal\log(\mathbf{T}_g^\intercal)\right) = 0.$

Proof: We can decouple this term into the translation and rotation terms enclosed in (15):

$$\begin{aligned}
&\mathrm{Tr}\left(\mathbf{X}_g^h(\mathbf{X}_g^h)^\intercal\log(\mathbf{T}_g^\intercal)\right) \\
&= \mathrm{Tr}\left((\mathbf{X}_g^\intercal, \mathbf{1}_{M_g})^\intercal(\mathbf{X}_g^\intercal, \mathbf{1}_{M_g})\log(\mathbf{T}_g^\intercal)\right) \\
&= \mathrm{Tr}\left(\begin{pmatrix} \mathbf{X}_g\mathbf{X}_g^\intercal & M_g\bar{\mathbf{x}}_g \\ M_g\bar{\mathbf{x}}_g^\intercal & M_g \end{pmatrix}\left(\begin{array}{c|c} \log(\mathbf{R}_g^\intercal) & \\ (\mathbf{t}')^\intercal & \mathbf{0}_{4\times 1} \end{array}\right)\right) \\
&= \mathrm{Tr}\left(\begin{array}{c|c} \mathbf{X}_g\mathbf{X}_g^\intercal\log(\mathbf{R}_g^\intercal) + M_g\bar{\mathbf{x}}_g(\mathbf{t}')^\intercal & \\ M_g\bar{\mathbf{x}}_g^\intercal\log(\mathbf{R}_g^\intercal) + M_g(\mathbf{t}')^\intercal & \mathbf{0}_{4\times 1} \end{array}\right) \\
&= \mathrm{Tr}\left(\mathbf{X}_g\mathbf{X}_g^\intercal\log(\mathbf{R}_g^\intercal)\right) + M_g\mathrm{Tr}\left(\bar{\mathbf{x}}_g\bar{\mathbf{y}}_g^\intercal\mathbf{V}_g^\intercal\ \right) \\
&= \mathrm{Tr}\left(\mathbf{X}_g\mathbf{X}_g^\intercal\log(\mathbf{R}_g^\intercal)\right) + M_g\mathrm{Tr}\left(\bar{\mathbf{x}}_g^\intercal\mathbf{V}_g\bar{\mathbf{y}}_g\ \right)
\end{aligned} \tag{18}$$

where we considered $\mathbf{t}' = \mathbf{V}_g\mathbf{t}_g = \mathbf{V}_g\bar{\mathbf{y}}_g$ and, in the last step, trace invariance with respect to matrix transposition and cyclic permutations. The second right hand side term in the last step of (18) is zero since frame of reference $\mathcal{F}_g(t)$ ensures $\bar{\mathbf{x}}_g = \mathbf{0}\,\forall t$ and the first term from the last step involves the trace of a symmetric matrix times a skew-symmetric matrix and thus is always zero. ■

Lemma III.2. $-\mathrm{Tr}\left(\mathbf{Y}_g\mathbf{X}_g^\intercal\log(\mathbf{R}_g^\intercal)\right) \leq 0\,\forall t$ *and* $\theta \in (-\pi, \pi)$.

Proof: Considering $\log(\mathbf{R}_g^\intercal) = -\log(\mathbf{R}_g)$ we use (16) to obtain:

$$\begin{aligned}
&\mathrm{Tr}\left(\mathbf{Y}_g\mathbf{X}_g^\intercal\log(\mathbf{R}_g)\right) \\
&\qquad = \frac{\theta}{2\sin(\theta)}\left(\left\langle\mathbf{R}_g^\intercal, \mathbf{Y}_g\mathbf{X}_g^\intercal\right\rangle_F - \left\langle\mathbf{R}_g, \mathbf{Y}_g\mathbf{X}_g^\intercal\right\rangle_F\right),
\end{aligned} \tag{19}$$

where $\langle\mathbf{A}, \mathbf{B}\rangle_F = \mathrm{Tr}(\mathbf{A}^\intercal\mathbf{B})$ is the Frobenius inner product. We define $\widetilde{\mathbf{Y}}_g = \mathbf{Y}_g - \bar{\mathbf{Y}}_g$ ($\bar{\mathbf{Y}}_g \in \mathbb{R}^{3\times M_g}$ stacks mean vector $\bar{\mathbf{y}}_g$). Matrix $\mathbf{R}_g(t, r^*(t))$ is obtained from the Procrustes analysis $\mathcal{P}(\mathbf{X}_g, \mathbf{Y}_g)$ in (1) through the singular value decomposition of matrix $\widetilde{\mathbf{Y}}_g\mathbf{X}_g^\intercal = \mathbf{Q}_g\mathbf{S}_g\mathbf{W}_g^\intercal$. Considering that $\bar{\mathbf{Y}}_g\mathbf{X}_g^\intercal = 0$ when $\bar{\mathbf{x}}_g = 0$, $\widetilde{\mathbf{Y}}_g\mathbf{X}_g^\intercal = \mathbf{Y}_g\mathbf{X}_g^\intercal = \mathbf{Q}_g\mathbf{S}_g\mathbf{W}_g^\intercal$. The optimal rotation (i.e. the minimiser of the Procrustes optimisation problem) can be computed as $\mathbf{R}_g = \mathbf{Q}_g\mathbf{W}_g^\intercal$. Pre-multiplying and pos-multiplying all four bracket terms in (19) by $\mathbf{Q}_g^\intercal$ and $\mathbf{W}_g$ respectively (both are unitary matrix, so the result is invariant to them), we obtain:

$$\begin{aligned}
&\frac{\theta}{2\sin(\theta)}\left(\left\langle\mathbf{Q}^\intercal\mathbf{R}_g^\intercal\mathbf{W}_g, \mathbf{S}_g\right\rangle_F - \langle\mathbf{I}_3, \mathbf{S}_g\rangle_F\right) \\
&\qquad = \frac{\theta}{2\sin(\theta)}\left(\mathrm{Tr}\left(\mathbf{W}_g^\intercal\mathbf{R}_g\mathbf{Q}_g\mathbf{S}_g\right) - \mathrm{Tr}\left(\mathbf{S}_g\right)\right),
\end{aligned} \tag{20}$$

Term $\frac{\theta}{2\sin(\theta)}$ is always positive for $\theta \in (-\pi, \pi)$ and, since diagonal matrix $\mathbf{S}_g$ contains all positive eigenvalues $\mathrm{Tr}(\mathbf{I}_3\mathbf{S}_g)$ is always equal or larger than $\mathrm{Tr}(\mathbf{W}_g^\intercal\mathbf{R}_g\mathbf{Q}_g\mathbf{S}_g)$ (note $\mathbf{W}_g^\intercal\mathbf{R}_g\mathbf{Q}_g \in SO(3)$). Therefore, (20) is $\leq 0\,\forall t$ and $\theta \in (-\pi, \pi)$. ■

Lemma III.3. $-M_g\mathrm{Tr}\left(\bar{\mathbf{y}}_g^\intercal\mathbf{V}_g\bar{\mathbf{y}}_g\right) \leq 0$, $\forall t$ *and* $\theta \in (-\pi, \pi)$.

Proof: The second term on the right hand side of the definition of $\mathbf{V}_g$ in (17) is a skew symmetric matrix and thus can be neglected ($\mathrm{Tr}(\mathbf{x}^\intercal\mathbf{A}\mathbf{x}) = \mathbf{0}$ for all $\mathbf{A}$ skew symmetric matrices). The third term on the right hand side in (17) when using (16) and applying Rodrigues' rotation formula to matrices $\mathbf{R}_g$ yields:

$$\frac{c}{\theta^2}(\log(\mathbf{R}_g))^2 = c\mathbf{K}^2, \text{ with } c(\theta) = 1 - \frac{\theta(1+\cos(\theta))}{2\sin(\theta)}, \quad (21)$$

where $\mathbf{K}(\theta)$ is the skew-symmetric matrix from Rodrigues formula and $c(\theta) \leq 1$ for $\theta \in (-\pi, \pi)$. With these considerations:

$$\begin{aligned} -M_g \mathrm{Tr}\left(\bar{\mathbf{y}}_g^\intercal \mathbf{V}_g \bar{\mathbf{y}}_g\right) &= -M_g \mathrm{Tr}\left(\bar{\mathbf{y}}_g^\intercal (\mathbf{I}_3 + c\mathbf{K}^2)\bar{\mathbf{y}}_g\right) \\ &= -M_g \mathrm{Tr}\left(\bar{\mathbf{y}}_g^\intercal \mathbf{I}_3 \bar{\mathbf{y}}_g\right) - M_g \mathrm{Tr}\left(\bar{\mathbf{y}}_g^\intercal c\mathbf{K}^2 \bar{\mathbf{y}}_g\right) \\ &= -M_g \mathrm{Tr}\left(\bar{\mathbf{y}}_g^\intercal \mathbf{I}_3 \bar{\mathbf{y}}_g\right) - M_g \mathrm{Tr}\left(\bar{\mathbf{y}}_g^\intercal \mathrm{diag}(-c, -c, 0)\bar{\mathbf{y}}_g\right) \\ &= -M_g \mathrm{Tr}\left(\bar{\mathbf{y}}_g^\intercal \mathrm{diag}(1-c, 1-c, 1)\bar{\mathbf{y}}_g\right), \end{aligned} \quad (22)$$

where in the third line we applied the fact that $\mathbf{K}^2$ is a symmetric matrix and the eigenvalues of a squared 3×3 skew-symmetric matrix are always $\{-1, -1, 0\}$. Note how (22) is always ≤ 0 for $\theta \in (-\pi, \pi)$ as $c < 1$ for $\theta \in (-\pi, \pi)$. ■

Theorem III.4. *The system error $e(t)$ (2) is locally asymptotically stable with control law (13). In particular, scalar function $\mathcal{V}(t) = \frac{1}{2}\|\mathbf{X}(t) - \mathbf{Y}\|_F^2$ is Lyapunov and presents a derivative $\dot{\mathcal{V}}_g \leq 0, \forall t$ and $\theta \in (-\pi, \pi)$.*

Proof: $\mathcal{V}(t) > 0, \forall \mathbf{X}(t) \neq \mathbf{Y}$ and $\mathcal{V}(t) = 0$ only when $\mathbf{X}(t) = \mathbf{Y}$. We consider that gripper-unaffected points present a zero derivative, i.e. $\mathrm{d}\mathbf{X}(t)/\mathrm{d}t = \mathbf{0}$ for $\mathbf{x}_m(t) \notin \Omega_g(t)$. We also consider that gripper-affected points' derivative is $\mathrm{d}\mathbf{X}_g^h/\mathrm{d}t = \log(\mathbf{T}_g(t, r^*(t))\mathbf{X}_g^h(t, r^*(t))$, i.e. deriving (14) as in (5). Therefore, the derivative of $\mathcal{V}(t)$ yields:

$$\dot{\mathcal{V}} = \frac{1}{G}\sum_g^G \dot{\mathcal{V}}_g, \quad (23)$$

where $\dot{\mathcal{V}} \leq 0$ if $\dot{\mathcal{V}}_g \leq 0$, $\forall g$. Each gripper-associated term $\dot{\mathcal{V}}_g$ is

$$\begin{aligned} \dot{\mathcal{V}}_g &= \mathrm{Tr}\left((\mathbf{X}_g^h - \mathbf{Y}_g^h)(\mathbf{X}_g^h)^\intercal \log(\mathbf{T}_g^\intercal)\right) \\ &= \mathrm{Tr}\left(\mathbf{X}_g^h(\mathbf{X}_g^h)^\intercal \log(\mathbf{T}_g^\intercal)\right) - \mathrm{Tr}\left(\mathbf{Y}_g^h(\mathbf{X}_g^h)^\intercal \log(\mathbf{T}_g^\intercal)\right). \end{aligned} \quad (24)$$

Applying lemma III.1, the first right hand side in (24) is zero and, therefore, $\dot{\mathcal{V}}_g$ yields:

$$\begin{aligned} \dot{\mathcal{V}}_g &= -\mathrm{Tr}\left(\mathbf{Y}_g^h(\mathbf{X}_g^h)^\intercal \log(\mathbf{T}_g^\intercal)\right) \\ &= -\mathrm{Tr}\left(\begin{pmatrix} \mathbf{Y}_g\mathbf{X}_g^\intercal & M_g\bar{\mathbf{y}}_g \\ M_g\bar{\mathbf{x}}_g^\intercal & M_g \end{pmatrix}\left(\begin{array}{c|c} \log(\mathbf{R}_g^\intercal) & \\ (\mathbf{t}')^\intercal & \mathbf{0}_{4\times1} \end{array}\right)\right) \\ &= -\mathrm{Tr}\left(\begin{array}{c|c} \mathbf{Y}_g\mathbf{X}_g^\intercal \log(\mathbf{R}_g^\intercal) + M_g\bar{\mathbf{y}}_g(\mathbf{t}')^\intercal & \\ M_g\bar{\mathbf{x}}_g \log(\mathbf{R}_g^\intercal) + M_g(\mathbf{t}')^\intercal & \mathbf{0}_{4\times1} \end{array}\right) \\ &= -\mathrm{Tr}\left(\mathbf{Y}_g\mathbf{X}_g^\intercal \log(\mathbf{R}_g^\intercal)\right) - M_g\mathrm{Tr}\left(\bar{\mathbf{y}}_g^\intercal \mathbf{V}_g\bar{\mathbf{y}}_g\right), \end{aligned} \quad (25)$$

where, for the second right hand side term, we considered $\mathbf{t}' = \mathbf{V}_g\mathbf{t}_g = \mathbf{V}_g\bar{\mathbf{y}}_g$ and trace invariance with respect to matrix transposition and cyclic permutations. Considering lemmas III.2 and III.3, both right hand side terms in (25) are ≤ 0, $\forall t$ and $\theta \in (-\pi, \pi)$ and thus the proof is completed. Note that computation of r^* relies in the relaxed assumption of LRB. For this reason, the stability analysis remains local since there is no guarantee for a global LRB. ■

Remark III.5. *Note that, as deformable objects constitute under-actuated systems, the local stability in Theorem III.4 does not guarantee a zero-valued error residual. However, it does ensure convergence of the closed-loop system towards a state with a zero error derivative.*

IV. Results

A. Simulations

We performed several simulations using the ARAP [8] deformation model. The simulation results are presented in Fig. 2. Within this figure, each column presents the results of a simulation. We normalised the colour map depicting values of $q(t, r)$ so that negative values are blue, positive values are yellow and zero (or close to zero) values are green. One intuition for interpreting surface $q(t, r)$ is to think of each scale $r \in [r_0, R(t)]$ (at a given instant t) as an available choice of an action to be performed under the relaxed LRB assumption. The value of $q(t, r)$ (11) constitutes the error derivative estimation that each action will cause. Our method seeks large error reduction by *choosing* actions $r^*(t)$ (12) that present large negative values of $q(t, r^*(t))$.

The first simulation in Fig. 2 constitutes a challenging case given its anti-symmetry. This is specially noticeable in the last plot where surface $q(t, r)$ presents estimations of positive error derivatives at large scales right at the initial configuration ($t = 0$, yellow tones around $r = 7$ [cm]). In other simulations, positive derivative estimations begin to appear later during the control process. Nonetheless, our initial choice of $r^*(t_0)$ ensures $q(t, r^*) < 0$ and the control task is performed properly.

The simulation in the second column in Fig. 2 constitutes a paradigmatic example of how our multi-scale analysis works. Note how our strategy is able to prioritise larger elements of the object (i.e., larger $r^*(t)$ values) when their contribution to the error term is large. The gripper positioned at the small appendix will collaborate in reducing the main beam error. Whenever the main beam error is reduced, the focus shall return to the reduction of the appendix' error (note the scale transition of $r^*(t)$ between $t = 5$ and $t = 10$). The third column in Fig. 2 involves a pure-twist. This can be challenging given singularities of the opposed rotation actions. However, actions are defined symmetrically and the error is smoothly reduced. Fourth column in Fig. 2 constitutes an example of pure bending process. Additional simulations are shown in the **attached video**.

B. Experiments

We present several experiments (Fig. 3) with deformation cases analogous to those shown in the simulations. Our setup consists of two ABB IRB120 industrial robots (equipped with pneumatic grippers) and an Intel Realsense D415 RGB-D camera that provides the depth information we use for the 3D analysis. The objects to be handled are single-coloured foam cut-outs that favour colour-based segmentation. Recall that our method is designed for relatively rigid objects that

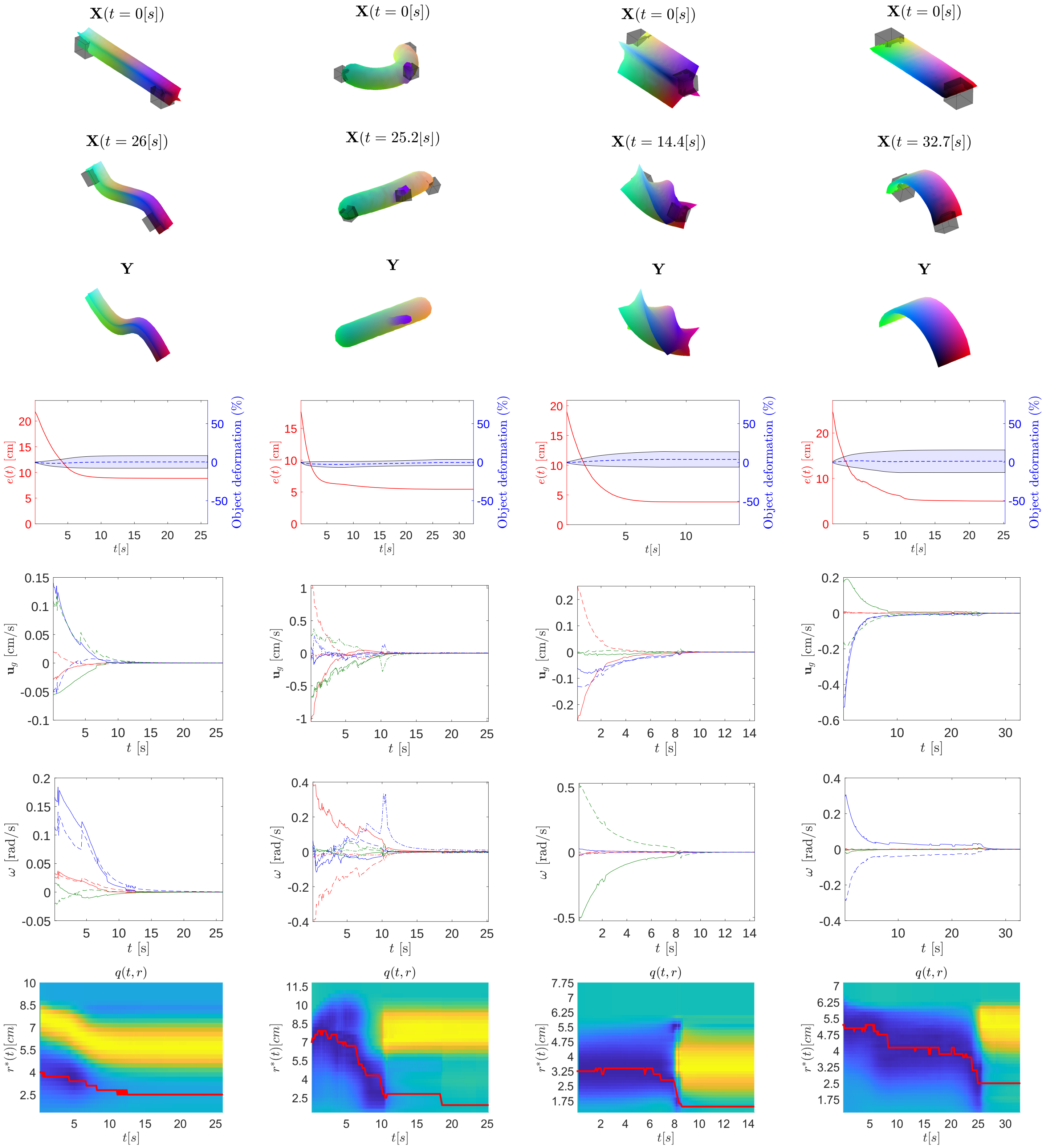

Fig. 2: Four examples, one per column, are presented. First three rows show the initial ($\mathbf{X}(t_0)$), final ($\mathbf{X}(t)$) and target ($\mathbf{Y}$) shapes. Each gripper is represented with a grey cube. On the object surface, the surface point matching is represented with a colour map. The fourth row contains the shape error plot (2) and a distribution of the object's relative deformation (positive values when the object stretches and negative values if the object is being compressed). The fifth and sixth rows are the action plots of the grippers: translation vector (i.e. $\mathbf{u}_g$) and rotation (Euler angles w) components. The RGB colour code indicates x, y, z components for translation and roll, pitch, yaw components for rotation. In the action-related plots (i.e. fifth and sixth rows), each line-style refers to a particular gripper. The last row represents surface $q(t, r)$ and the evolution of $r^*(t)$ through time (red line).

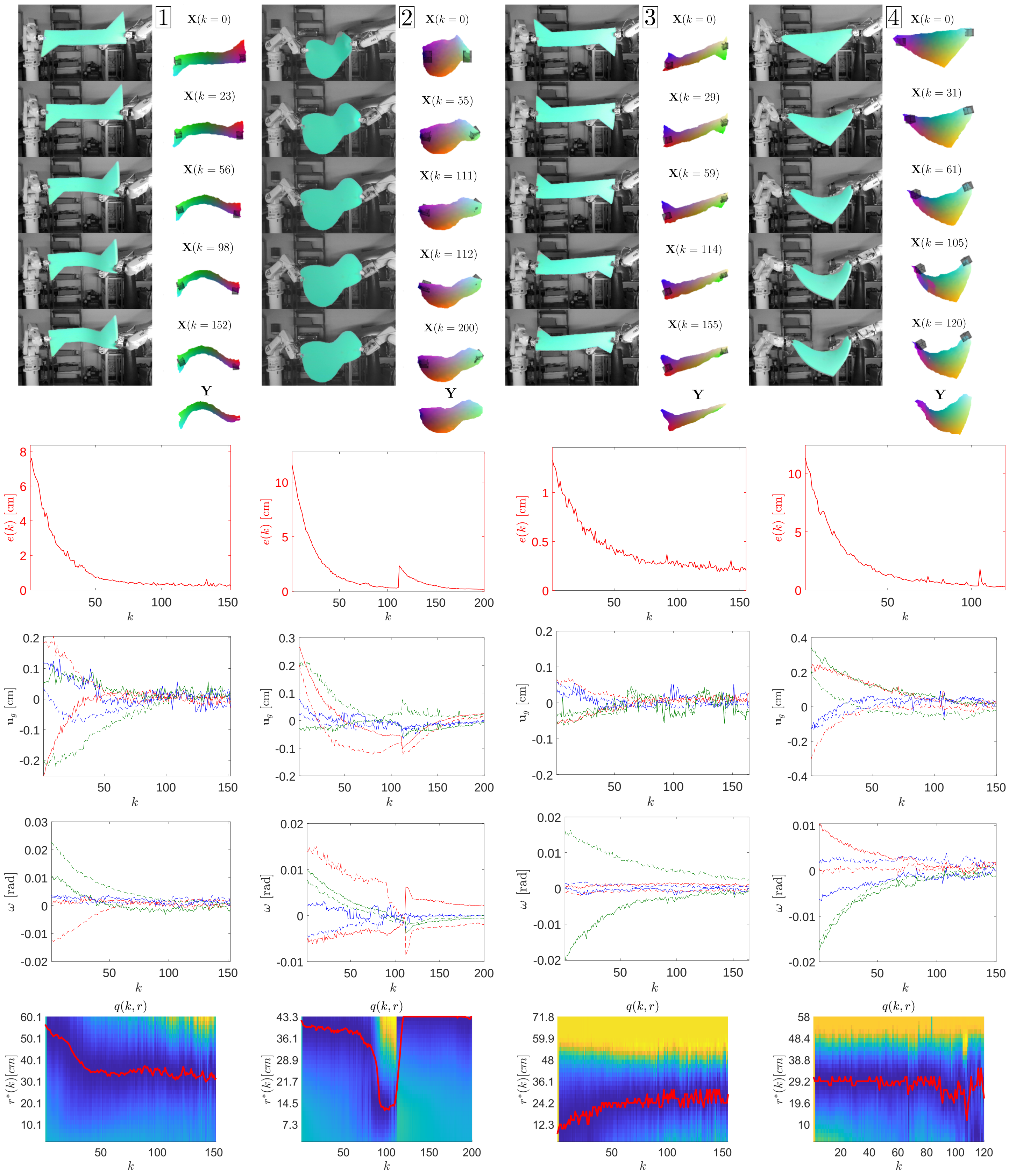

Fig. 3: Four experiments, from left to right, that constitute analogous deformation cases to those of Fig. 2. First block shows several snapshots of the experiments and the target shape ($\mathbf{Y}$). Each gripper is represented with a grey cube. The surface point matching is represented with a colour map. Next rows contain the shape error, the translation action vector $\mathbf{u}_g$, and rotation action components (Euler angles ω). In the action plots, each line-style refers to a particular gripper, and the RGB colour code indicates x, y, z components. Last row represents surface $q(k,r)$ and the evolution of $r^*(k)$ (red line).

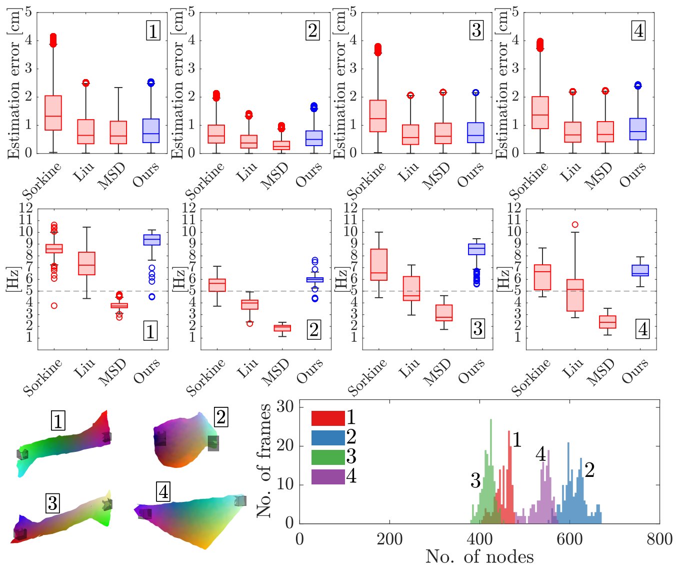

Fig. 4: The figure presents a comparison of our proposed shape control model against two geometry-based deformation models [8] and [32], and against as a mass-spring-damper (MSD) physical-based model. On the top row, for each of the four experiments, the models' estimation error is presented. The estimation error is determined by evaluating the distance distribution between predicted and measured node positions across iterations. On the second row, the time cost of each process is represented in terms of processing frequency.

align with our local-rigidity behaviour assumption. Therefore, we cannot guarantee its proper performance with softer objects, such as clothes. The analysis is carried out in Matlab and the communication with the robots is via TCP/IP. To avoid dropping below the process frequency of 5 Hz with the previously mentioned set-up, we propose discretising r to a set of around 20 scale values. Note that, given the analysis in Appendix A, $q(r, t)$ in (12) is continuous differentiable. Therefore, using discrete points to approximate partial derivatives of $q(r, t)$ by means of classic methods (e.g., polynomial regression) holds mathematical significance. Regarding the code implementation, experiments were conducted on an Intel(R) Core(TM) i7-8565U CPU with 1.99 GHz and 16 GB of RAM [1].

All shape control problems tackled in the experiments involve 3D surfaces. Target shapes have been pre-defined with robot configurations that ensure that the shape is achievable; this ensures, for example, that the robot is not required to leave its workspace. The setup, the object-segmentation and several relevant time instants are shown in Fig. 3. For each of the four experiments several acquired RGB images are shown grayscale along with the object segmentation highlighted in blue. The object's 3D points (corresponding to each time instant) are shown from a slightly elevated viewpoint that provides a better understanding of the 3D configuration of the object. The grippers are represented with gray cubes, and the target shape $\mathbf{Y}$ with the surface mapping (colour map on the object) is shown at the bottom of the sequence. The **attached video** presents several additional perspectives that provide a better insight of the 3D configurations of the objects.

The first experiment in Fig. 3 involves an anti-symmetric bending process along with a light twist. A trend-change can be observed on the actions after $k = 20$, specially on the y and z components of $\mathbf{u}_g$ (green and blue) and the y components of ω (green). This is a consequence of the evolution of $r^*(k)$ during the first 50 iterations: our system focuses first on the bending process as it is responsible for larger error reduction and then tackles the twisting process.

The second experiment consists of unfolding an object from a bent to a straightened configuration. This shape control problem is particularly challenging as, when approaching the target shape (around $k = 110$) the object buckles thus breaking the assumption of small deformations between iterations and generating unreliable estimations of $q(k, r)$. The buckling produces a global shape change on the object and thus $r^*(k)$ undergoes an abrupt change towards global scale values (i.e., $r^*(t) = R(t)$). Even so, the system shows robustness and manages to continue converging. The buckling process can be better perceived in Fig. 3 and in the top view of the experiments video: between $k = 111$ and $k = 112$ the object suddenly changes its curvature with respect to the camera's front view.

The third experiment in Fig. 3 involves a pure-twisting process. As in the analogue simulation, the alignment of the rotation axes can be problematic, as opposite rotation actions are coupled. Our control system generates symmetric rotation actions that evenly deform both sides of the object. Optimal scale $r^*(t)$ increases as the object increases its rigidity due to the accumulated torsion: the object's response to pure rotation at larger scales takes some iterations. This is not the case in the analogous simulation, as the ARAP deformation model prioritises homogeneous deformation along the whole object from the initial time instant. The fourth experiment involves the 3D bending of a triangle shape. The shape control task is properly managed. However, it is worth mentioning a small error increase at $k = 105$. This is due to a discontinuity in the surface mapping that infringes the assumption of surface continuity. This lack of map-continuity can be appreciated in the last column of Fig. 3, where a colour map discontinuity can be observed on the left side of the triangle ($k = 105$). As a result, peaks (outliers) around $k = 105$ can be observed in all the plots of the fourth column in Fig. 3.

Figure 4 shows the analysis of our proposed Procrustes-based deformation model regarding estimation error, computation frequency, and volume of analysed data. The estimation error is defined by comparing predicted and measured node positions during iterations. We compared our model against geometry-based deformation models from [8] (used in [6] [7]) and [32], as well as a mass-spring-damper (MSD) physical-based model. Our model competes well with these three models, achieving a desirable balance between accuracy and computation time-cost. In addition, unlike the time cost of the other three models, our process' time cost also encompasses

[1] Available demo code for the method is provided in https://github.com/nachocz/Multi-scale-Procrustes-based-3D-shape-control

the cost of computing the control actions. The favourable balance between model accuracy and low computation time cost makes our model an attractive choice for shape control. Note that these comparisons do not aim to constitute conventional bench-marking: some aspects, such as the type of mesh or the set maximum number of optimisation iterations, may affect the performance of different deformation models in distinct manners. This makes it difficult to make fair comparisons. However, seeking fairness in the comparisons, we imposed a maximum limit of 20 iterations on the optimisation process for all of the compared models. This approach allows for a reasonable evaluation of model performance in shape control, despite the inherent complexities involved in comparing diverse methods.

V. Conclusions

We developed a novel 3D shape control strategy that does not rely on physical models but rather on a multi-scale shape analysis that allows to prioritise those parts of the objects that not only contribute more to the error metric but also can be influenced by the grippers to an estimated extent. We provided theoretical analysis of local asymptotic stability of the control system and performed several simulations and experiments that validate the system showing robustness and proper error convergence. To the best of our knowledge this is the first 3D shape control framework involving multi-scale geometric analysis of texture-less objects. Regarding future work, an interesting research line would be decoupling the scale at which each gripper is acting, i.e. independently defining an r^* for each gripper and thus (most likely) achieving more error reduction in certain cases. Note that this is nontrivial as the dimensionality of the problem increases by G (i.e., $r_1 \times ... \times r_G$).

Appendix

A. *Differentiability of* $\mathbf{T}_g$ *with respect to* r

In this appendix we study continuity and differentiability of $\mathbf{T}_g$ (i.e. of $\mathbf{R}_g, \mathbf{t}_g$) with respect to r. Variations in r imply a smooth variation on the domains Ω_g being considered for the Procrustes analysis (domains are enlarged or reduced by boundary $\partial\Omega_g$). Regarding the continuous Procrustes problem definition in [30], we can define our analogous space-continuous squared Procrustes residual for $\mathbf{X}_g$ and $\mathbf{Y}_g$ as:

$$d^2_{\mathcal{P}}(\mathbf{X}_g, \mathbf{Y}_g) = \min_{\mathbf{R}_g, \mathbf{t}_g} \left(\int_{\Omega_g} \left\| \mathbf{R}_g \mathbf{x} + \mathbf{t}_g - \mathbf{y} \right\|^2 \mathrm{d}\Omega \right), \quad (26)$$

where $\mathbf{x}$ and $\mathbf{y}$ represent mapped points of the shapes' surface manifolds. For now on, as $\mathbf{x}$ are mapped to $\mathbf{y}$ with area preserving maps [24], we will use $\mathrm{d}\Omega_g$ to refer to the differential elements on both the current and the target shape manifolds. The optimal translation, in this continuous formulation, is

$$\mathbf{t}_g = \frac{\int_{\Omega_g} \mathbf{y} \mathrm{d}\Omega_g}{\int_{\Omega_g} \mathrm{d}\Omega_g} - \frac{\int_{\Omega_g} \mathbf{R}_g \mathbf{x} \mathrm{d}\Omega_g}{\int_{\Omega_g} \mathrm{d}\Omega_g} = \frac{\int_{\Omega_g} (\mathbf{y} - \mathbf{R}_g \mathbf{x}) \mathrm{d}\Omega_g}{\int_{\Omega_g} \mathrm{d}\Omega_g}, \quad (27)$$

and can be differentiated with respect to r to obtain:

$$\begin{aligned} \frac{\partial}{\partial r} \mathbf{t}_g &= \frac{\partial}{\partial r} \left(\frac{\int_{\Omega_g} (\mathbf{y} - \mathbf{R}_g \mathbf{x}) \mathrm{d}\Omega_g}{\int_{\Omega_g} \mathrm{d}\Omega_g} \right) \\ &= \frac{\partial}{\partial r} \left(\frac{1}{A} \right) \int_{\Omega_g} (\mathbf{y} - \mathbf{R}_g \mathbf{x}) \mathrm{d}\Omega_g + \frac{1}{A} \frac{\partial}{\partial r} \left(\int_{\Omega_g} (\mathbf{y} - \mathbf{R}_g \mathbf{x}) \mathrm{d}\Omega_g \right) \\ &= -\frac{1}{A^2} \frac{\partial A}{\partial r} \int_{\Omega_g} (\mathbf{y} - \mathbf{R}_g \mathbf{x}) \mathrm{d}\Omega_g \\ &+ \frac{1}{A} \left(\int_{\Omega_g} \frac{\partial \mathbf{R}_g}{\partial r} \mathbf{x} \mathrm{d}\Omega_g + \int_{\partial\Omega_g} (\mathbf{y} - \mathbf{R}_g \mathbf{x}) \mathrm{d}(\partial\Omega_g) \right), \quad (28) \end{aligned}$$

where $A(r)$ is the area of the open domain Ω_g and, in the last step, we applied the Leibniz's integral rule. Note that (28) depends on the existence of $\frac{\partial \mathbf{R}_g}{\partial r}$. The optimal Procrustes rotation component $\mathbf{R}_g$ can be obtained from:

$$\mathbf{R}_g = \sqrt{\mathbf{M}_g \mathbf{M}_g^{\intercal}} \, \mathbf{M}_g^{-1}, \quad (29)$$

where $\mathbf{M}_g(r) \in \mathbb{R}^{3\times 3}$ is equal to:

$$\mathbf{M}_g(r) = \int_{\Omega_g} (\mathbf{x} - \bar{\mathbf{x}}_g)(\mathbf{y} - \bar{\mathbf{y}}_g)^{\intercal} \, \mathrm{d}\Omega_g. \quad (30)$$

In this continuous formulation $\bar{\mathbf{x}}_g = \int_{\Omega_g} \mathbf{x} \mathrm{d}\Omega_g / \int_{\Omega_g} \mathrm{d}\Omega_g$ and $\bar{\mathbf{y}}_g = \int_{\Omega_g} \mathbf{y} \mathrm{d}\Omega_g / \int_{\Omega_g} \mathrm{d}\Omega_g$. Equation (29), and therefore $\mathbf{R}_g$, is continuous differentiable with respect to r given two conditions: $\mathbf{M}_g$ being continuous differentiable and existence of $\mathbf{M}_g^{-1}$ for $r \in [r_0, R(t)]$. $\mathbf{M}_g(r)$ is differentiable with respect to $r, \forall\, r > 0$:

$$\begin{aligned} \frac{\partial}{\partial r} \mathbf{M}_g = &\int_{\Omega_g} \frac{\partial}{\partial r} \left((\mathbf{x} - \bar{\mathbf{x}}_g)(\mathbf{y} - \bar{\mathbf{y}}_g)^{\intercal} \right) \mathrm{d}\Omega_g \\ &+ \int_{\partial\Omega_g} (\mathbf{x} - \bar{\mathbf{x}}_g)(\mathbf{y} - \bar{\mathbf{y}}_g)^{\intercal} \, \mathrm{d}(\partial\Omega_g). \quad (31) \end{aligned}$$

On the other hand, $\mathbf{M}_g^{-1}$ exists if there exist at least three non-aligned points $\mathbf{x}$ and three non-aligned points $\mathbf{y}$ in the domains Ω_g of each manifold. This condition is certainly achieved for any 3D object (planar or volumetric) and $r > 0$.

References

[1] R. Herguedas, G. López-Nicolás, R. Aragüés, and C. Sagüés. Survey on multi-robot manipulation of deformable objects. In *24th IEEE International Conference on Emerging Technologies and Factory Automation*, pages 977–984, 2019.

[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*, 37(7):688–716, 2018.

[3] H. Yin, A. Varava, and D. Kragic. Modeling, learning, perception, and control methods for deformable object manipulation. *Science Robotics*, 6(54):eabd8803, 2021.

[4] D. Berenson. Manipulation of deformable objects without modeling and simulating deformation. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 4525–4532, 2013.

[5] D. Navarro-Alarcón, Y. Liu, J.G. Romero, and P. Li. Model-free visually servoed deformation control of elastic objects by robot manipulators. *IEEE Transactions on Robotics*, 29(6):1457–1468, 2013.

[6] M. Shetab-Bushehri, M. Aranda, Y. Mezouar, and E. Ozgur. As-rigid-as-possible shape servoing. *IEEE Robotics and Automation Letters*, 7(2):3898–3905, 2022.

[7] M. Shetab-Bushehri, M. Aranda, Y. Mezouar, and E. Ozgur. Lattice-based shape tracking and servoing of elastic objects. *arXiv preprint arXiv:2209.01832*, 2022.

[8] O. Sorkine and M. Alexa. As-rigid-as-possible surface modeling. In *Symposium on Geometry processing*, volume 4, pages 109–116, 2007.

[9] D. Navarro-Alarcon and Y.H. Liu. Fourier-based shape servoing: a new feedback method to actively deform soft objects into desired 2-D image contours. *IEEE Transactions on Robotics*, 34(1):272–279, 2017.

[10] J. Zhu, B. Navarro, P. Fraisse, A. Crosnier, and A. Cherubini. Dual-arm robotic manipulation of flexible cables. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 479–484, 2018.

[11] A. Sintov, S. Macenski, A. Borum, and T. Bretl. Motion planning for dual-arm manipulation of elastic rods. *IEEE Robotics and Automation Letters*, 5(4):6065–6072, 2020.

[12] R. Lagneau, A. Krupa, and M. Marchal. Automatic shape control of deformable wires based on model-free visual servoing. *IEEE Robotics and Automation Letters*, 5(4):5252–5259, 2020.

[13] N. Lv, J. Liu, and Y. Jia. Dynamic modeling and control of deformable linear objects for single-arm and dual-arm robot manipulations. *IEEE Transactions on Robotics*, 38(4):2341–2353, 2022.

[14] D. Navarro-Alarcon, H.M. Yip, Z. Wang, Y.H. Liu, F. Zhong, T. Zhang, and P. Li. Automatic 3-D manipulation of soft objects by robotic arms with an adaptive deformation model. *IEEE Transactions on Robotics*, 32(2):429–441, 2016.

[15] J. Zhu, D. Navarro-Alarcon, R. Passama, and A. Cherubini. Vision-based manipulation of deformable and rigid objects using subspace projections of 2-D contours. *Robotics and Autonomous Systems*, 142:103798, 2021.

[16] M. Aranda, J.A. Corrales Ramon, Y. Mezouar, A. Bartoli, and E. Özgür. Monocular visual shape tracking and servoing for isometrically deforming objects. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 7542–7549, 2020.

[17] J. Qi, G. Ma, J. Zhu, P. Zhou, Y. Lyu, H. Zhang, and D. Navarro-Alarcon. Contour moments based manipulation of composite rigid-deformable objects with finite time model estimation and shape/position control. *IEEE/ASME Transactions on Mechatronics*, 27(5):2985–2996, 2022.

[18] Z. Hu, P. Sun, and J. Pan. Three-dimensional deformable object manipulation using fast online Gaussian process regression. *IEEE Robotics and Automation Letters*, 3(2):979–986, 2018.

[19] G. López-Nicolás, R. Herguedas, M. Aranda, and Y. Mezouar. Simultaneous shape control and transport with multiple robots. In *IEEE International Conference on Robotic Computing*, pages 218–225, 2020.

[20] A. Cherubini, V. Ortenzi, A. Cosgun, R. Lee, and P. Corke. Model-free vision-based shaping of deformable plastic materials. *The International Journal of Robotics Research*, 39(14):1739–1759, 2020.

[21] J. Zhu, C. Dune, M. Aranda, Y. Mezouar, J.A. Corrales, P. Gil, and G. López-Nicolás. Editorial: Robotic handling of deformable objects. *IEEE Robotics and Automation Letters*, 7(3):8257–8259, 2022.

[22] L. Han, H. Wang, Z. Liu, W. Chen, and X. Zhang. Vision-based cutting control of deformable objects with surface tracking. *IEEE/ASME Transactions on Mechatronics*, 26(4):2016–2026, 2020.

[23] X. Li, X. Su, and Y.H. Liu. Vision-based robotic manipulation of flexible PCBs. *IEEE/ASME Transactions on Mechatronics*, 23(6):2739–2749, 2018.

[24] S. Melzi, J. Ren, E. Rodola, A. Sharma, P. Wonka, and M. Ovsjanikov. Zoomout: Spectral upsampling for efficient shape correspondence. *ACM Transactions on Graphics*, 38(6):1–14, 2019.

[25] J. Zhu, B. Navarro, R. Passama, P. Fraisse, A. Crosnier, and A. Cherubini. Robotic manipulation planning for shaping deformable linear objects with environmental contacts. *IEEE Robotics and Automation Letters*, 5(1):16–23, 2020.

[26] S. Huo, A. Duan, C. Li, P. Zhou, W. Ma, H. Wang, and D. Navarro-Alarcon. Keypoint-based planar bimanual shaping of deformable linear objects under environmental constraints with hierarchical action framework. *IEEE Robotics and Automation Letters*, 7(2):5222–5229, 2022.

[27] T. Matsuno, D. Tamaki, F. Arai, and T. Fukuda. Manipulation of deformable linear objects using knot invariants to classify the object condition based on image sensor information. *IEEE/ASME Transactions on Mechatronics*, 11(4):401–408, 2006.

[28] I. Cuiral-Zueco and G. López-Nicolás. Multi-scale Laplacian-based FMM for shape control. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 3792–3797, 2021.

[29] Z. Hu, T. Han, P. Sun, J. Pan, and D. Manocha. 3-D deformable object manipulation using deep neural networks. *IEEE Robotics and Automation Letters*, 4(4):4255–4261, 2019.

[30] R. Al-Aifari, I. Daubechies, and Y. Lipman. Continuous procrustes distance between two surfaces. *Communications on Pure and Applied Mathematics*, 66(6):934–964, 2013.

[31] R. M. Murray, Z. Li, and S. S. Sastry. *A mathematical introduction to robotic manipulation*. CRC press, 2017.

[32] L. Liu, L. Zhang, Y. Xu, C. Gotsman, and S.J. Gortler. A local/global approach to mesh parameterization. In *Computer graphics forum*, volume 27, pages 1495–1504. Wiley Online Library, 2008.