Controlling the Shape of Deformable Linear Objects in 3D with a Simple Geometric Model

Miguel Burgh-Oliván
Inst. de Inv. en Ing. de Aragón (I3A)
Universidad de Zaragoza
Zaragoza, Spain
Email: mburgh@unizar.es

Miguel Aranda
Inst. de Inv. en Ing. de Aragón (I3A)
Universidad de Zaragoza
Zaragoza, Spain
Email: miguel.aranda@unizar.es

Gonzalo López-Nicolás
Inst. de Inv. en Ing. de Aragón (I3A)
Universidad de Zaragoza
Zaragoza, Spain
Email: gonlopez@unizar.es

***Abstract*—This paper addresses the robotic manipulation of deformable linear objects (DLOs) in 3D space, which is a complex problem with relevant applications in, e.g., industrial, agricultural, or medical domains. We propose a simple geometric model of elastic deformation tailored to this problem, and exploit this model to derive a quasi-static deformation Jacobian mapping robot motions to changes in the DLO's shape. We then propose a control law, based on this Jacobian, to drive multiple points on the DLO toward target positions. The described approach extends prior work restricted to 2D scenarios, and is capable of controlling the DLO's shape in 3D and with six-degrees-of-freedom gripper motions. The main advantage of this approach is its simplicity, as it does not require training, simulation, or full perception of the object's shape. We present validation results from both simulations and real-world experiments.**

***Index Terms*—Robotics, Automatic manipulation, Deformable objects.**

I. Introduction

In our daily lives, we easily manipulate deformable linear objects (DLOs) such as ropes, cables, plant stems, or wires. However, manipulating these objects with robotic systems is a major challenge due to their complex deformation dynamics. A growing number of research studies have been addressing the robotic manipulation of DLOs in recent years, stimulated by numerous applications in industry (routing wires), medical surgery (inserting needles and controlling catheters) and agriculture (fruit harvesting) [1], [2].

The particular shape and dynamic characteristics of DLOs, which distinguish them from other types of deformable objects, have been exploited to propose manipulation approaches based on well-known mechanical models such as Kirchhoff rods [3], [4] or Cosserat rods [5]. In the work [6], a more general manipulation planning framework was proposed incorporating the effect of gravity. The study [7] developed a comprehensive dynamic model of DLOs and addressed their deployment on planar workspaces with one or two robotic arms. Several works have addressed manipulation tasks by exploiting mechanical deformation models expressed via the finite element method (FEM) [8]–[10], or via alternatives such as the reproducing kernel particle method [11] or mass-spring-damper models [12], [13]. In the specific case of DLO manipulation, the work in [14] exploited an efficient FEM model for quasi-static deformation control for objects assumed to lie in a 2D workspace.

In most existing works addressing DLO shape control, the manipulation task proceeds within a feedback loop relying on sensor data about the actual state of the object, where vision is the most commonly used sensing modality. Certain approaches use the sensor data acquired during the task to construct, via online estimation techniques, a local model relating incremental manipulation actions and incremental changes in the features that are used to represent the object's shape. This type of approach was applied for various types of features and objects in [15], for DLOs in 2D in [16], [17] and for small 3D DLO deformations in [18]. A related technique was used in the study [19], which targeted DLO manipulation in agricultural scenarios and proposed adjusting the parameters of a deformation model during the task within an adaptive control framework.

The representational power of artificial neural networks has also been exploited in a number of approaches. The authors of [20] used a multilayer neural network to model the mapping between end-effector actions and object shape changes. In [21], an offline model of a DLO based on graph neural networks was employed, together with online refinement with a residual model to better handle model inaccuracies. A graph neural network was also used in [22], where a cable installation task was demonstrated. In [23], DLO dynamics were modeled with a radial-basis-function neural network, learned offline and adapted online, within an approach capable of versatile 3D manipulation. Meanwhile, the works [24], [25] exploited reinforcement learning for DLO shape control.

In this work, we consider the task of controlling the shape of a DLO via manipulation by robots. The task's goal is to drive several points on the object toward prescribed target positions, relying on visual perception of the positions of these points. The setup is illustrated in Fig. 1. Our core assumption is that the DLO's deformation can be represented with sufficient accuracy by an as-similar-as-possible (ASAP) geometric model [26]–[28], chosen as an approximation of

This work was supported by the Interreg Sudoe Programme, ERDF through the project REMAIN - S1/1.1/E0111, by MCIN/AEI/10.13039/501100011033, ERDF A way of making Europe, and the European Union NextGenerationEU/PRTR through the projects PID2021-124137OB-I00 and TED2021-130224B-I00, and by the Government of Aragón through the group T45_23R.

This is the final author version of an article published in the proceedings of the IEEE 2025 ETFA conference
The publisher's version and the citation information are available at https://doi.org/10.1109/ETFA65518.2025.11205790

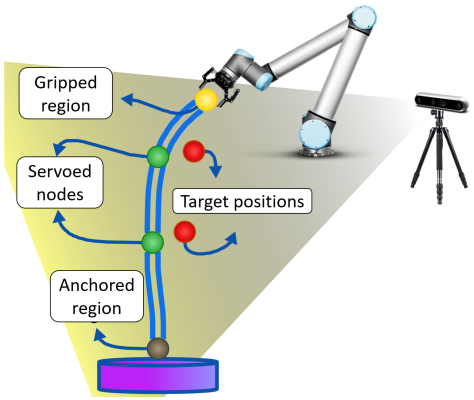

Fig. 1: Representation of the considered system. The addressed task consists in manipulating the DLO using the robot so that the servoed nodes reach their target positions. A vision sensor used to detect the positions of the servoed nodes is depicted.

the as-rigid-as-possible (ARAP) [29]–[31] geometric model of elastic deformation. Based on this assumption, we define an ASAP energy tailored to our scenario and derive from this energy a quasi-static deformation Jacobian and a control law for the robotic grippers. This Jacobian is a constant matrix, and its computation only requires choosing a discretization (number of nodes) to represent the DLO. We demonstrate the capabilities of this approach with simulations and experiments.

The contributions of this paper are:

- Our scheme extends the one in [28], which controlled DLO deformations in 2D and used gripper motions with three degrees of freedom (DOF), to 3D deformations and 6-DOF gripper motions. Many other works assume the DLO lies in a 2D workspace [7], [14], [16], [17], [21], [22], [28], [32]. Here, we handle the more complex case of DLOs deforming in 3D.
- As we rely on a constant Jacobian, we avoid the need for online estimations or online deformation model computations that are used in other approaches [14]–[19], [31]. While such approaches are interesting due to their adaptability, they can be sensitive to noise and initialization/excitation conditions. We do not need to use mechanical parameters, in contrast to [4]–[6], [14], and we do not require training either offline or online, differently from [20]–[25]. Hence, our approach has advantages in terms of simplicity, while remaining effective.
- The approach we propose is formulated using geometric analysis relying on an established deformation modeling framework (ASAP), and we do not use the perception of the object's full shape during the control task. These traits distinguish our approach from the solutions in [32]–[34].

II. Problem definition

The system we consider is visualized in Fig. 1. Next, we describe the elements of the system, our assumptions about them, and the task we solve.

- The manipulated object is a DLO (e.g., a flexible rod, a tube, a cable, a wire, or a plant stem). The object's material composition is unknown, but its deformation behavior is assumed to be mostly homogeneous along its length; i.e., the object has no articulations or regions with significantly different stiffness. The geometry of the object at its rest state is a straight line. The object deforms elastically; i.e., it tends to return to its rest shape when forces stop being applied on it. The manipulation is quasi-static and the object's shape is stable at every time instant. The changes in shape are continuous; e.g., buckling effects are disregarded. The effect of gravity is not modeled explicitly. The deformation is always locally small, meaning that there is never an extreme deformation at any point along the DLO. The global deformation of the full object is not constrained and, hence, can potentially be large.
- The object is gripped with a gripping configuration that fully constrains its shape. While the approach we will propose can be used with multiple movable grippers, we concentrate on the case where one end of the DLO is fixed (e.g., held by a static gripper, or fixed to the ground) and the other end is held by a movable robotic gripper at the end-effector of a robot arm. The 6-DOF velocity of this gripper at every instant can be fixed by a robot controller. The knowledge of which parts of the object are being gripped or fixed is available beforehand. These parts are the same during the full task. The parts not being gripped or fixed are free to move, i.e., they are not subjected to any external forces or contacts in the environment.
- The task is to control the gripper so that the 3D positions of several points on the object, called servoed nodes, move toward prescribed target positions. The target positions are reachable, i.e., there is a feasible configuration of the object where the servoed nodes are in those positions. The 3D positions of the servoed nodes, measured by a static vision sensor, are known during the task's execution. The transformation between the sensor coordinate frame and the robot coordinate frame is known. The kinematic model of the robot arm is also known.

III. DLO deformation energy model in 3D

The DLO is represented discretely, by a set of n consecutive nodes indexed from 1 to n. The model is illustrated in Fig. 2. We define a reference configuration representing the rest shape of the object, where the nodes are evenly spaced and they form a straight line. We denote the positions of the nodes in the rest shape by column vectors $\mathbf{c}_i \in \mathbb{R}^3$ and we denote the positions of the nodes on the object by column vectors $\mathbf{q}_i \in \mathbb{R}^3$, for $i = 1, 2, \ldots, n$. Without loss of generality, the positions are expressed in the reference frame of the static sensor. We denote the vector encoding the full object's shape by $\mathbf{q} = [\mathbf{q}_1^\intercal, \mathbf{q}_2^\intercal, \ldots, \mathbf{q}_n^\intercal]^\intercal \in \mathbb{R}^{3n}$. Our model divides the object into overlapping local regions, where each region is a set of three consecutive nodes, i.e., a *triad*. In order to cover the n nodes that form the object, there are $n-2$ triads, T_l, indexed

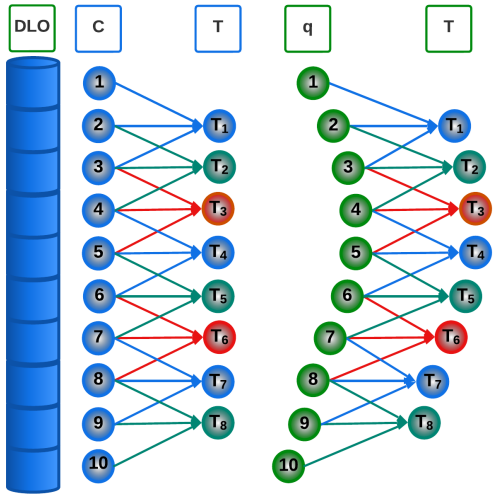

Fig. 2: Illustration of the model of the DLO. The object is discretized in n nodes ($n = 10$ in this depiction). In the rest configuration $\mathbf{c}$, shown in blue, the nodes are evenly spaced and form a straight line. The positions of these nodes on the actual DLO, $\mathbf{q}$, are represented in green. For defining the deformation energy, the nodes are grouped in $n-2$ triads, T_l for $l = 1, \ldots, n-2$. The different triads are represented with different colors, for clarity.

from $l = 1$ to $l = n-2$, where the triad T_l comprises nodes l, $l+1$, and $l+2$. We model the internal deformation energy of the DLO as an ASAP energy. We choose to use this modeling for two reasons:

1) ARAP is a well established method for modeling elastic deformations [30]. It is defined based on an optimal rigid transformation for every local region. ASAP can be defined in an analogous way, but using an optimal similarity transformation instead. The two transformations are equal (i.e., the scale of the similarity is 1) when there is no local deformation, and they are close to each other when the local deformation is small [28]. Then, our hypothesis is that ASAP can be a proper method for modeling, in an approximate manner and in the context of the addressed task, small elastic local deformations.
2) The ASAP method (unlike the ARAP method) leads to a *constant* deformation Jacobian relating motions of gripped and servoed nodes of the DLO. This greatly simplifies the control task and allows us to avoid identifications of mechanical parameters, online estimations, training, or sensing of the DLO's full shape.

We take the 2D scenario considered in [28] as our starting point. In the following developments, we use $(\cdot)'$ to denote the variables of the 2D case. The work in [28] used an ASAP deformation energy in 2D, $E' \in \mathbb{R}_{\geq 0}$, defined as the sum of an energy $E'_l \in \mathbb{R}_{\geq 0}$ for every triad l. For the case of a DLO with $n-2$ triads, this has the expression

$$E' = \sum_{l=1}^{n-2} E'_l, \;\; E'_l = \frac{1}{2} \sum_{i=l}^{l+2} \| (\mathbf{q}'_i - \mathbf{q}'_{0_l}) - \mathbf{H}'_l (\mathbf{c}'_i - \mathbf{c}'_{0_l}) \|^2. \quad (1)$$

Here, $\mathbf{q}'_{0_l}$ and $\mathbf{c}'_{0_l}$ denote the centroids for the nodes of the triad l in the current configuration and in the rest configuration, respectively. $\mathbf{H}'_l \in \mathbb{R}^{2\times 2}$ is the least-squares similarity transformation, comprising rotation and scaling, that relates the current and rest positions for triad l. Hence, E'_l expresses the difference between two triad shapes: the current one, and the rest one transformed by an optimal similarity. This is why E' is an ASAP energy. By using the analytical expression of $\mathbf{H}'_l$, one can write E' as

$$E' = -\frac{1}{2} \mathbf{q}'^{\intercal} \mathbf{A}' \mathbf{q}'. \quad (2)$$

For the specific case where the rest shape is a straight line of evenly-spaced nodes, which is the case treated in this paper, the matrix $\mathbf{A}'$ has the expression

$$\mathbf{A}' = \mathbf{B} \otimes \mathbf{I}_2 \in \mathbb{R}^{2n\times 2n}, \;\; \text{with } \mathbf{B} = \sum_{l=1}^{n-2} \mathbf{P}_l^{\intercal} \mathbf{A}_t \mathbf{P}_l \in \mathbb{R}^{n\times n}. \quad (3)$$

Here, $\otimes$ denotes the Kronecker product and $\mathbf{I}_2$ denotes the 2×2 identity matrix, the matrix $\mathbf{A}_t \in \mathbb{R}^{3\times 3}$ is

$$\mathbf{A}_t = \begin{bmatrix} -\frac{1}{6} & \frac{1}{3} & -\frac{1}{6} \\ \frac{1}{3} & -\frac{2}{3} & \frac{1}{3} \\ -\frac{1}{6} & \frac{1}{3} & -\frac{1}{6} \end{bmatrix}, \quad (4)$$

and the matrices $\mathbf{P}_l \in \mathbb{R}^{3\times n}$ for $l = 1, 2, \ldots, n-2$ are selector matrices. Denoting by $[\cdot, \cdot]$ the entries of a matrix, these matrices are such that $\mathbf{P}_l[1, l]$, $\mathbf{P}_l[2, l+1]$, and $\mathbf{P}_l[3, l+2]$ are equal to one, and all other entries are zero.

Notice that the structure of $\mathbf{A}'$, given by the matrix $\mathbf{B}$, is separable and identical for each of the two spatial coordinates. From this observation, we propose here using the same matrix in 3D, by simply extending the dimensionality. Concretely, using the 3×3 identity matrix, $\mathbf{I}_3$, we define the matrix

$$\mathbf{A} = \mathbf{B} \otimes \mathbf{I}_3 \in \mathbb{R}^{3n\times 3n}. \quad (5)$$

This can also be interpreted as a combination of matrices $\mathbf{A}'$ for the projections in the three planes xy, yz, xz, because

$$\mathbf{A} = \mathbf{B} \otimes \mathbf{I}_3 = \frac{1}{2}\Big(\mathbf{B} \otimes \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix} + \mathbf{B} \otimes \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} + \mathbf{B} \otimes \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} \Big). \quad (6)$$

Then, the ASAP energy we propose in the 3D case is

$$E = -\frac{1}{2} \mathbf{q}^{\intercal} \mathbf{A} \mathbf{q}. \quad (7)$$

Interestingly, E does not depend on the specific positions of the rest configuration points $\mathbf{c}_i$. This is a consequence of using an ASAP model; if we scale, rotate, and translate the points $\mathbf{c}_i$, the optimal similarity for every triad correspondingly undergoes scaling, rotation, and translation in such a way that E_l, and hence E, remain unchanged. The model we propose aims for simplicity, and does not include a representation of torsion in the DLO. This simplification makes our model more suitable for tasks where torsion is undesired or small.

IV. Deformation Jacobian definition

In this section, we use the energy E in (7) to define virtual forces as linear functions of $\mathbf{q}$ and ultimately derive a *constant* matrix, obtained directly from the matrix $\mathbf{A}$, mapping between the displacements of the part of the object being gripped and the part of the object we want to control. From this, we will define our control law. Interpreting E as a potential energy associated with the object's deformation, we define $\mathbf{f} = [\mathbf{f}_1^\intercal, \mathbf{f}_2^\intercal, \ldots, \mathbf{f}_n^\intercal]^\intercal \in \mathbb{R}^{3n}$ as the vector of ASAP forces; i.e., the virtual forces at the nodes due to the energy E, where $\mathbf{f}_i \in \mathbb{R}^3$ is the force at the node i. Since $\mathbf{A}_t$ is symmetric, $\mathbf{A}$ is symmetric too, and the forces are, from (7), as follows:

$$\mathbf{f} = -\frac{\partial E}{\partial \mathbf{q}} = \mathbf{A}\mathbf{q}. \tag{8}$$

We now derive a quasi-static deformation Jacobian by using a strategy based on grouping the nodes in sets according to their role in the addressed task. Related strategies have been used in the domain of deformable object manipulation; e.g., in [12]. Here, we specifically propose an approach analogous to those employed in [14], [28]. We group the nodes into n_g gripped nodes, n_s servoed nodes, and n_f free nodes, such that $n_g + n_s + n_f = n$. Note that the set of gripped nodes includes all nodes constrained externally; e.g., the nodes held by a robotic gripper and the nodes fixed to the ground. The positions of the nodes in these sets are grouped, respectively, in $\mathbf{q}_g \in \mathbb{R}^{3n_g}$, $\mathbf{q}_s \in \mathbb{R}^{3n_s}$, and $\mathbf{q}_f \in \mathbb{R}^{3n_f}$. We divide $\mathbf{A}$ and $\mathbf{f}$ in blocks accordingly with the same corresponding subscripts. Taking the time derivative of (8), we can obtain

$$\begin{bmatrix} \mathbf{A}_{gg} & \mathbf{A}_{gs} & \mathbf{A}_{gf} \\ \mathbf{A}_{sg} & \mathbf{A}_{ss} & \mathbf{A}_{sf} \\ \mathbf{A}_{fg} & \mathbf{A}_{fs} & \mathbf{A}_{ff} \end{bmatrix} \begin{bmatrix} \dot{\mathbf{q}}_g \\ \dot{\mathbf{q}}_s \\ \dot{\mathbf{q}}_f \end{bmatrix} = \begin{bmatrix} \dot{\mathbf{f}}_g \\ \dot{\mathbf{f}}_s \\ \dot{\mathbf{f}}_f \end{bmatrix}. \tag{9}$$

The object's shape is always stable by assumption. Therefore, the total force (i.e., sum of the ASAP force $\mathbf{f}_i$ and any external force) at every node i is always zero. Thus, the time derivative is zero as well. Calling the external force on node i $\mathbf{f}_{e,i}$, we then have $\dot{\mathbf{f}}_i + \dot{\mathbf{f}}_{e,i} = [0,0,0]^\intercal$ for $i = 1, 2, \ldots, n$. For non-gripped nodes, i.e., servoed nodes and free nodes, the external force is zero or constant (e.g., the gravity force). Therefore, for every non-gripped node, we have $\dot{\mathbf{f}}_{e,i} = [0,0,0]^\intercal$ and, hence, $\dot{\mathbf{f}}_i = [0,0,0]^\intercal$. Using this fact in (9), we obtain

$$\begin{bmatrix} \mathbf{A}_{sg} & \mathbf{A}_{ss} & \mathbf{A}_{sf} \\ \mathbf{A}_{fg} & \mathbf{A}_{fs} & \mathbf{A}_{ff} \end{bmatrix} \begin{bmatrix} \dot{\mathbf{q}}_g \\ \dot{\mathbf{q}}_s \\ \dot{\mathbf{q}}_f \end{bmatrix} = \begin{bmatrix} \mathbf{0} \\ \mathbf{0} \end{bmatrix}. \tag{10}$$

For these two equation sets, we can solve for $\dot{\mathbf{q}}_f$ in the second and then substitute into the first, obtaining

$$\dot{\mathbf{q}}_s = \mathbf{J}_{sg}\dot{\mathbf{q}}_g, \tag{11}$$

where

$$\mathbf{J}_{sg} = -\Big(\mathbf{A}_{ss} - \mathbf{A}_{sf}\mathbf{A}_{ff}^{-1}\mathbf{A}_{fs}\Big)^{-1}\Big(\mathbf{A}_{sg} - \mathbf{A}_{sf}\mathbf{A}_{ff}^{-1}\mathbf{A}_{fg}\Big). \tag{12}$$

$\mathbf{J}_{sg} \in \mathbb{R}^{3n_s \times 3n_g}$ is a constant deformation Jacobian matrix that maps between the displacements of the gripped and servoed nodes of the DLO. We assume that the inverse matrices in (11) exist, which is the case in situations where the shape of the object is completely constrained by the gripped nodes.

V. Control law

Let us denote the target positions for the servoed nodes by $\mathbf{q}_d \in \mathbb{R}^{3n_s}$, so that the goal is to achieve $\mathbf{q}_s = \mathbf{q}_d$. We define the error of the control task, i.e., the servoing error, as

$$\mathbf{e}_s = \mathbf{q}_s - \mathbf{q}_d. \tag{13}$$

The idea of our control law is to exploit (11) in order to apply velocities to the gripped nodes so that they make $\mathbf{e}_s$ zero, i.e., they move the servoed nodes toward their target positions. In the real system, velocities are not applied directly to each separate gripped node; instead, multiple gripped nodes are moved jointly by a robotic gripper that can rotate and translate with six degrees of freedom. Therefore, we define a gripper control law next. For simplicity of the exposition and to fit the actual setup we implemented, we consider a single gripper grasping two DLO nodes. The extension to multiple grippers and different numbers of gripped nodes is straightforward.

We assume that the gripper is grasping two adjacent nodes of the DLO with indices g_1 and g_2. We denote by $\mathbf{h} \in \mathbb{R}^3$ the position of the center of the gripper. The vectors from this center to the gripped nodes are

$$\mathbf{r}_i = [r_{ix}, r_{iy}, r_{iz}]^\intercal = \mathbf{q}_{g_i} - \mathbf{h} \quad \text{for } i = 1, 2. \tag{14}$$

We denote by $\mathbf{v}_h = [v_x, v_y, v_z, \omega_x, \omega_y, \omega_z]^\intercal \in \mathbb{R}^6$ the 6-DOF velocity vector of the gripper, where v denotes the translational (linear) velocities and ω denotes the rotational (angular) velocities. The rotational action is applied about the center of the gripper, and x, y, z are the axes of the reference frame where the positions $\mathbf{q}_i$ are expressed. Then, following [31], we obtain the matrices that map the gripper's 6-DOF motion to the motion of each of the two gripped nodes as

$$\mathbf{J}_{ih} = \begin{bmatrix} 1 & 0 & 0 & 0 & r_{iz} & -r_{iy} \\ 0 & 1 & 0 & -r_{iz} & 0 & r_{ix} \\ 0 & 0 & 1 & r_{iy} & -r_{ix} & 0 \end{bmatrix} \quad \text{for } i = 1, 2, \tag{15}$$

in such a way that $\dot{\mathbf{q}}_{g_i} = \mathbf{J}_{ih}\mathbf{v}_h$. We stack these matrices in a matrix $\mathbf{J}_{gh}$, having size $3n_g \times 6$ and containing zeros in the positions corresponding to gripped nodes that are not grasped by the gripper (e.g., nodes fixed on the ground, or grasped by another gripper). For example, if g_1 and g_2 are the indices corresponding to the first two positions in $\mathbf{q}_g$, then $\mathbf{J}_{gh} = [\mathbf{J}_{1h}^\intercal, \mathbf{J}_{2h}^\intercal, \mathbf{0}_{6\times(3n_g-6)}]^\intercal$. We define $\mathbf{J}_{sh} = \mathbf{J}_{sg}\mathbf{J}_{gh}$, which is the Jacobian that links the motions of the gripper and the servoed nodes. Then, the control law we propose is

$$\mathbf{v}_h = -k_h \mathbf{J}_{sh}^{+}\mathbf{e}_s, \tag{16}$$

where $(\cdot)^+$ denotes the Moore-Penrose inverse and $k_h \in \mathbb{R}_{>0}$ is a control gain. Alternatively, it is possible to use a gain matrix instead of k_h; e.g., a diagonal matrix of positive gains

can be employed in order to give different weights to the translational and rotational velocities.

From classical Jacobian-based control analysis, e.g., in [35], the system defined by the error state $\mathbf{e}_s$ can remain stable with the proposed control law if the computed Jacobian approximates sufficiently well the dynamics of the actual DLO. When the number of servoed nodes grows and the system becomes underactuated, stability guarantees become local. In various scenarios, it is reasonable to control, as proposed in this paper, only a reduced number of servoed nodes. This includes cases where the positions of these nodes essentially determine the shape of the full DLO, or where the goal of the task (e.g., assembly) is the placement of some specific parts of the DLO.

VI. Experimental validation

In this section, we present results from our validation of the proposed DLO manipulation approach in simulation and in real-world experiments. In all the tests we present, we use target positions of the servoed nodes that are reachable by the DLO. We define target positions requiring DLO deformations in 3D. In the presented results, the servoed nodes are numbered in an ascending order from the fixed DLO end; i.e., servoed node 1 is the one closest to the fixed end in the rest configuration. Videos of our tests are available at https://www.youtube.com/playlist?list=PLp_Aa-IfaBfFFeoUTUEwtjoO2XRBjil4A.

A. Simulations

We implement the proposed control approach in MATLAB. We use a virtual DLO which we simulate using the ARAP method in [30] to create a behavior similar to a thin elastic tube with a circular cross-section. A free-floating gripper (i.e., not attached to a robot arm) grasping two nodes is simulated at one end of the DLO. The two nodes at the other end of the DLO are fixed to the ground. To move the gripper, we apply the control law (16). We use the gain $k_h = 0.2$ and simulate the control law with a time step of 0.02 s. We do not simulate a vision sensor, and we run the control law using the object's centerline positions expressed in the simulation's global coordinate frame.

In the first test we report (see Fig. 3), the DLO has a length of 0.8 m, and we use $n = 30$ nodes to discretize it in our control approach. It can be seen that the control system drives the two servoed nodes toward their target positions. In the remaining tests, the DLO has a length of 1 m, and we use $n = 50$ nodes. This way, we illustrate that our approach can handle different DLO lengths and values of n. In particular, n should be chosen high enough to correctly capture the deformability of the object. In Fig. 4, we illustrate a test with two servoed nodes where a considerable deformation of the DLO is required. As can be seen, our approach is capable of handling this case.

We then present results of tests with three servoed nodes. This is generally a more complex scenario, as the number of degrees of freedom to be controlled is higher. On the other hand, using three (or more) servoed nodes is interesting because it can allow for finer control of the full shape of the

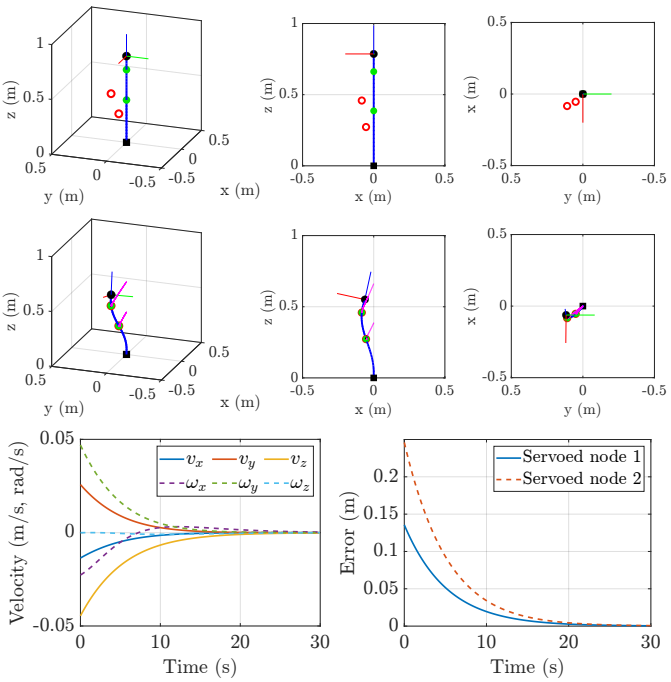

Fig. 3: Results of a simulation test with two servoed nodes. The target (red hollow circles) and current (green solid circles) positions of the servoed nodes are shown. The gripper's position is marked by a black circle and its orientation is marked by three (in red, green, blue colors) line segments. The fixed region at the bottom of the DLO is marked by a black square. The centerline of the DLO is plotted as a blue solid line. The top row shows three views of the initial configuration. The second row shows the same views for the final configuration, and includes the paths followed by the servoed nodes as magenta solid lines. The bottom row shows time plots of the gripper's linear and angular velocities (left) and of the Euclidean norm of the 3D servoing error corresponding to each servoed node (right).

DLO, compared to using two servoed nodes. In Fig. 5, we show the results of a successful test with three servoed nodes. For the test illustrated in Fig. 6, one can see that the system converges to a configuration with a non-negligible residual error. In general, this may occur in cases where completing the task involves complex deformations or large motions of the servoed nodes. One possible direction to follow in order to solve this issue would be to use deformation planning. For example, one could define a path of target positions to be achieved sequentially, so that at all times the target positions are close to the current servoed node positions. Another possible way to alleviate this limitation is by using multiple grippers.

B. Real-world experiments

In this section, we describe an evaluation of the proposed approach in a real-world scenario. We use a UR10 robotic arm. For visual perception purposes, we use ArUco markers,

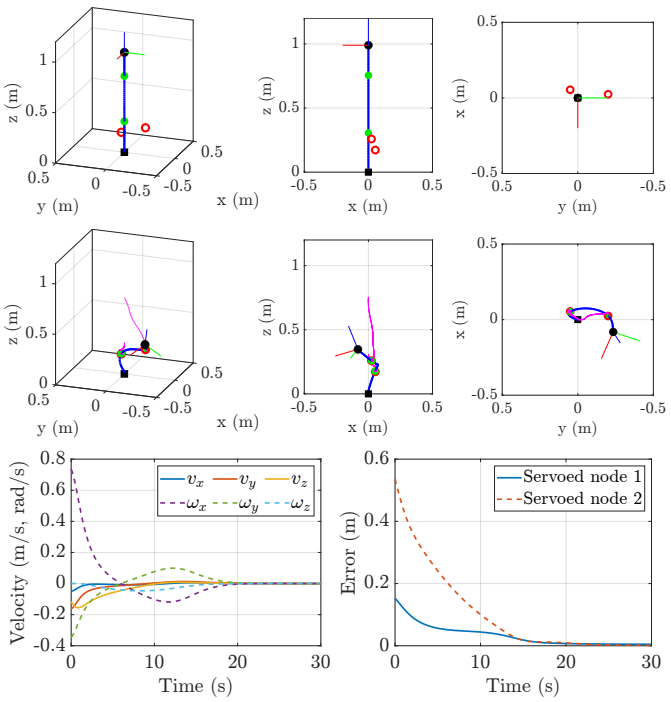

Fig. 4: Results of a simulation test with two servoed nodes and a large deformation. The interpretation of the figure's contents is the same as in Fig. 3.

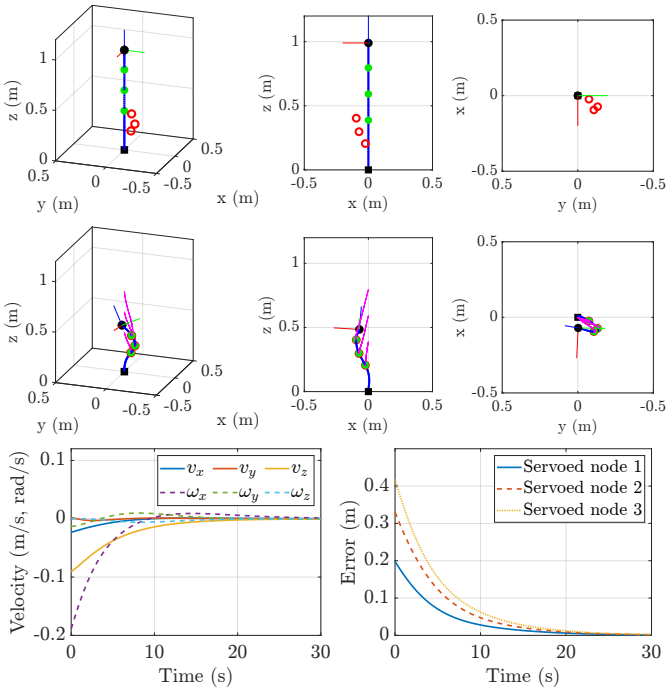

Fig. 5: Results of a simulation test with three servoed nodes. The interpretation of the figure's contents is the same as in Fig. 3.

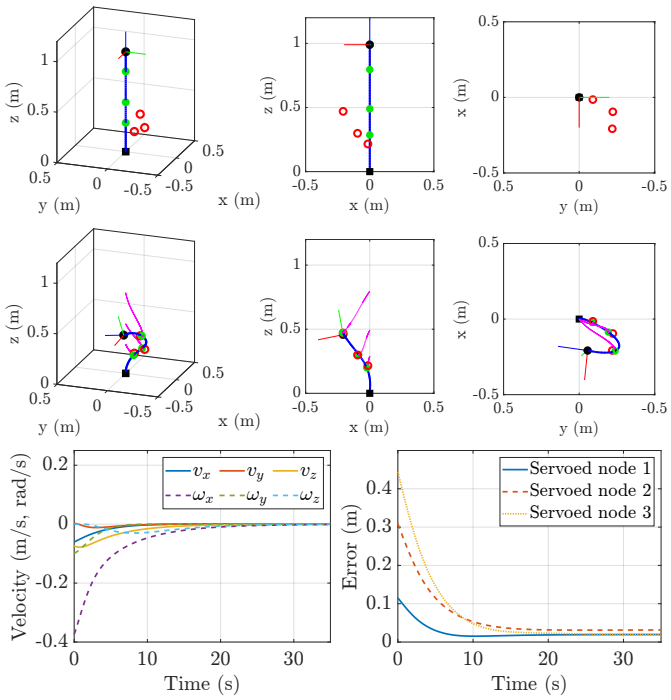

Fig. 6: Results of a simulation test with three servoed nodes with a noticeable final residual error. The interpretation of the figure's contents is the same as in Fig. 3.

with their centers being placed at the servoed nodes. An Intel RealSense D435 camera is used to detect the 3D positions of the markers' centers during the task. We implement the full control system in ROS. We present results from three tests with two different objects. For both objects, we choose $n = 50$. We use the camera coordinate frame, with axes x, y, and z in the frontward (i.e., depth), leftward, and upward directions, respectively. In the first test, illustrated in Fig. 7, we use an object made of synthetic latex having a length of 50 cm and a circular cross-section with a diameter of 3.5 cm. We use two servoed nodes and only apply translational velocities to the gripper in this test. The performance is satisfactory, illustrating that our approach is capable of solving tasks that do not require gripper rotation.

The second and third tests are done with a different DLO made of foam, having a length of 50 cm and a circular cross-section with a diameter of 2.8 cm. In the second test, illustrated in Fig. 8, we control one single servoed node, which is the simplest scenario for our approach. As seen in the plots, our approach solves this case well, with a gripper motion consisting mostly of translation. The third test, illustrated in Fig. 9, is with three servoed nodes. Reaching the target positions in this test involves a large deformation of the DLO. It can be seen that the control system is capable of reducing the errors to small final values. The performance in these real-world tests is affected by perturbations due to various causes; these include the inaccuracy of the used 3D positions of the servoed nodes, particularly in the spatial coordinate corresponding to the depth in the camera frame.

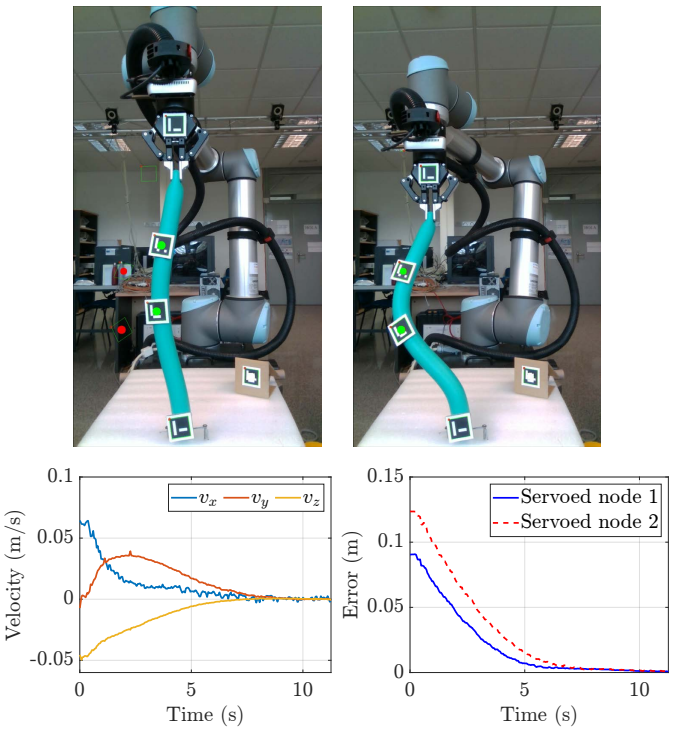

Fig. 7: Results of the experiment with two servoed nodes and the first DLO. First row: initial (left) and final (right) images of the system, captured by the camera used in the control approach. The current positions of the servoed nodes are marked as overlaid green circles, and their target positions are marked as overlaid red circles. Second row: time plots of the control law velocities (left) and of the Euclidean norm of the 3D servoing error corresponding to each of the servoed nodes (right).

The presented validation shows that the proposed approach can complete DLO shape control tasks in 3D while having simpler requirements in terms of modeling and perception than other alternative approaches.

VII. Conclusion

This paper addressed the development of an approach for the robotic manipulation of DLOs in 3D environments. The main idea was to use a simple geometric model of elastic deformation; for this, we extended the 2D control method in [28] to encode the deformation behavior in 3D space. The proposed control approach is simple to define and it enables DLO manipulation without the need to simulate or sense the entire object during execution. Its capability to successfully control the shape of DLOs was demonstrated with simulation examples and real-world experiments. As the approach is formulated on the basis of a Jacobian which is a simplified model and which only represents the deformation behavior under small incremental actions, it may fail when complex and large deformations are required. In addition, it is an approach designed for controlling the shape of predominantly elastic DLOs, and it is not well suited for other types of DLOs or

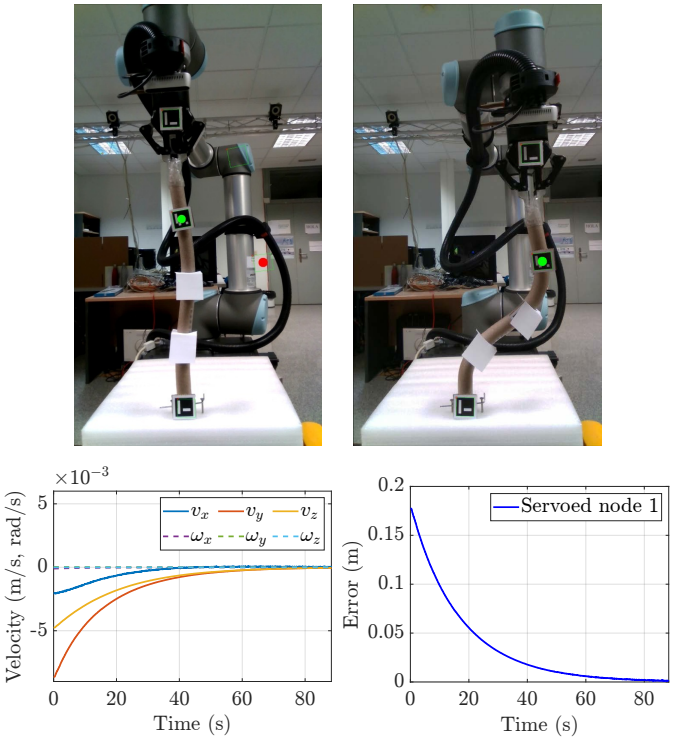

Fig. 8: Results of the experiment with one servoed node and the second DLO. The interpretation of the figure's contents is the same as in Fig. 7.

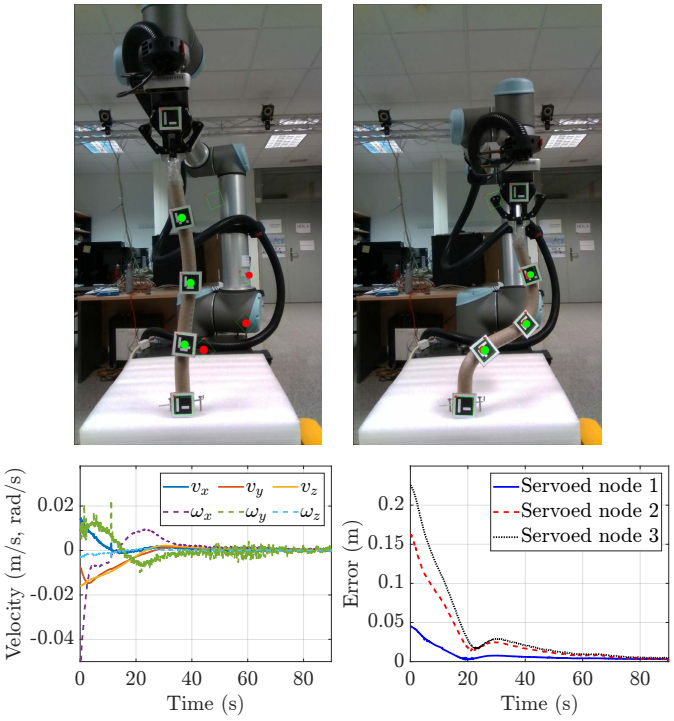

Fig. 9: Results of the experiment with three servoed nodes and the second DLO. The interpretation of the figure's contents is the same as in Fig. 7.

for more general types of deformable objects. Future work directions include addressing these issues, conducting further validations of the approach in more diverse conditions, and targeting broader application scenarios, such as human-robot collaborative manipulation.

REFERENCES

[1] H. Yin, A. Varava, and D. Kragic, "Modeling, learning, perception, and control methods for deformable object manipulation," *Science Robotics*, vol. 6, no. 54, p. eabd8803, 2021.

[2] J. Zhu, A. Cherubini, C. Dune, D. Navarro-Alarcon, F. Alambeigi, D. Berenson, F. Ficuciello, K. Harada, J. Kober, X. Li, J. Pan, W. Yuan, and M. Gienger, "Challenges and outlook in robotic manipulation of deformable objects," *IEEE Robotics & Automation Magazine*, vol. 29, no. 3, pp. 67–77, 2022.

[3] T. Bretl and Z. McCarthy, "Quasi-static manipulation of a Kirchhoff elastic rod based on a geometric analysis of equilibrium configurations," *The International Journal of Robotics Research*, vol. 33, no. 1, pp. 48–68, 2014.

[4] A. Sintov, S. Macenski, A. Borum, and T. Bretl, "Motion planning for dual-arm manipulation of elastic rods," *IEEE Robotics and Automation Letters*, vol. 5, no. 4, pp. 6065–6072, 2020.

[5] A. Artinian, F. B. Amar, and V. Perdereau, "Closed-loop shape control of deformable linear objects based on Cosserat model," *IEEE Robotics and Automation Letters*, vol. 9, no. 10, pp. 8746–8753, 2024.

[6] S. Wu, J. Zhang, and D. Wu, "An optimization-based motion planner for dual-arm manipulation of the soft deformable linear objects with nonnegligible gravity," *Advanced Engineering Informatics*, vol. 62, p. 102874, 2024.

[7] 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*, vol. 38, no. 4, pp. 2341–2353, 2022.

[8] S. Duenser, J. Bern, R. Poranne, and S. Coros, "Interactive robotic manipulation of elastic objects," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2018, pp. 3476–3481.

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

[10] M. O. Fonkoua, F. Chaumette, and A. Krupa, "Deformation control of a 3D soft object using RGB-D visual servoing and FEM-based dynamic model," *IEEE Robotics and Automation Letters*, vol. 9, no. 8, pp. 6943–6950, 2024.

[11] J. Smolen and A. Patriciu, "Deformation planning for robotic soft tissue manipulation," in *2009 Second International Conference on Advances in Computer-Human Interactions*, 2009, pp. 199–204.

[12] T. Wada, S. Hirai, S. Kawamura, and N. Kamiji, "Robust manipulation of deformable objects by a simple PID feedback," in *2001 IEEE International Conference on Robotics and Automation (ICRA) (Cat. No.01CH37164)*, 2001, pp. 85–90 vol.1.

[13] J. Das and N. Sarkar, "Autonomous shape control of a deformable object by multiple manipulators," *Journal of Intelligent & Robotic Systems*, vol. 62, pp. 3–27, 2011.

[14] A. Koessler, N. R. Filella, B.-C. Bouzgarrou, L. Lequièvre, and J.-A. Corrales Ramon, "An efficient approach to closed-loop shape control of deformable objects using finite element models," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*, 2021, pp. 1637–1643.

[15] D. Navarro-Alarcon, Y.-H. Liu, J. Romero, and P. Li, "On the visual deformation servoing of compliant objects: uncalibrated control methods and experiments," *International Journal of Robotics Research*, vol. 33, no. 11, pp. 1462–1480, 2014.

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

[17] J. Qi, G. Ran, B. Wang, J. Liu, W. Ma, P. Zhou, and D. Navarro-Alarcon, "Adaptive shape servoing of elastic rods using parameterized regression features and auto-tuning motion controls," *IEEE Robotics and Automation Letters*, vol. 9, no. 2, pp. 1428–1435, 2024.

[18] R. Lagneau, A. Krupa, and M. Marchal, "Automatic shape control of deformable wires based on model-free visual servoing," *IEEE Robotics and Automation Letters*, vol. 5, no. 4, pp. 5252–5259, 2020.

[19] O. Aghajanzadeh, M. Shetab-Bushehri, M. Aranda, J. A. Corrales Ramon, C. Cariou, R. Lenain, and Y. Mezouar, "3-D shape control of deformable linear objects for branch handling using an adaptive Lyapunov-based scheme," *Computers and Electronics in Agriculture*, vol. 232, p. 109931, 2025.

[20] 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*, vol. 4, no. 4, pp. 4255–4261, 2019.

[21] C. Wang, Y. Zhang, X. Zhang, Z. Wu, X. Zhu, S. Jin, T. Tang, and M. Tomizuka, "Offline-online learning of deformation model for cable manipulation with graph neural networks," *IEEE Robotics and Automation Letters*, vol. 7, no. 2, pp. 5544–5551, 2022.

[22] B. Cao, X. Zang, X. Zhang, Z. Chen, S. Li, and J. Zhao, "Shape control of elastic deformable linear objects for robotic cable assembly," *Advanced Intelligent Systems*, vol. 6, no. 7, p. 2300835, 2024.

[23] M. Yu, K. Lv, H. Zhong, S. Song, and X. Li, "Global model learning for large deformation control of elastic deformable linear objects: An efficient and adaptive approach," *IEEE Transactions on Robotics*, vol. 39, no. 1, pp. 417–436, 2023.

[24] R. Laezza and Y. Karayiannidis, "Learning shape control of elastoplastic deformable linear objects," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*, 2021, pp. 4438–4444.

[25] M. Daniel, A. Magassouba, M. Aranda, L. Lequièvre, J. A. Corrales Ramón, R. Iglesias Rodriguez, and Y. Mezouar, "Multi actor-critic DDPG for robot action space decomposition: A framework to control large 3D deformation of soft linear objects," *IEEE Robotics and Automation Letters*, vol. 9, no. 2, pp. 1318–1325, 2024.

[26] R. Chen and C. Gotsman, "Generalized as-similar-as-possible warping with applications in digital photography," *Computer Graphics Forum*, vol. 35, no. 2, pp. 81–92, 2016.

[27] G.-X. Zhang, M.-M. Cheng, S.-M. Hu, and R. R. Martin, "A shape-preserving approach to image resizing," *Computer Graphics Forum*, vol. 28, no. 7, pp. 1897–1906, 2009.

[28] O. Aghajanzadeh, M. Aranda, G. López-Nicolás, R. Lenain, and Y. Mezouar, "An offline geometric model for controlling the shape of elastic linear objects," in *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2022, pp. 2175–2181.

[29] T. Igarashi, T. Moscovich, and J. F. Hughes, "As-rigid-as-possible shape manipulation," *ACM Transactions on Graphics*, vol. 24, no. 3, pp. 1134–1141, 2005.

[30] O. Sorkine and M. Alexa, "As-rigid-as-possible surface modeling," in *Eurographics Symposium on Geometry Processing*, 2007, pp. 109–116.

[31] M. Shetab-Bushehri, M. Aranda, Y. Mezouar, and E. Özgür, "Lattice-based shape tracking and servoing of elastic objects," *IEEE Transactions on Robotics*, vol. 40, pp. 364–381, 2024.

[32] K. Almaghout, A. Cherubini, and A. Klimchik, "Robotic co-manipulation of deformable linear objects for large deformation tasks," *Robotics and Autonomous Systems*, vol. 175, p. 104652, 2024.

[33] D. Berenson, "Manipulation of deformable objects without modeling and simulating deformation," in *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2013, pp. 4525–4532.

[34] M. Ruan, D. McConachie, and D. Berenson, "Accounting for directional rigidity and constraints in control for manipulation of deformable objects without physical simulation," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2018, pp. 512–519.

[35] F. Chaumette and S. Hutchinson, "Visual servo control. I. Basic approaches," *IEEE Robotics & Automation Magazine*, vol. 13, no. 4, pp. 82–90, 2006.