Optimal Cosserat-based deformation control for robotic manipulation of linear objects

Artinian Azad
ISIR
Sorbonne University
Paris, France
artinian@isir.upmc.fr

Huet Quentin
ISIR
Sorbonne University
Paris, France
huet@isir.upmc.fr

Ben Amar Faiz
ISIR
Sorbonne University
Paris, France
amar@isir.upmc.fr

Perdereau Véronique
ISIR
Sorbonne University
Paris, France
veronique.perdereau@isir.upmc.fr

Abstract—Deformable object manipulation is a challenging research subject that draws a growing interest in the robotic field as new methods to tackle this problem have emerged. So far, most of the proposed approaches in the literature focused solely on shape control. The strain applied to the object is disregarded, thus excluding a large part of industrial applications where fragile products are manipulated, like the demolding of rubber and plastic objects, or the handling of food. These applications require a tradeoff between accuracy and careful manipulation in order to preserve the manipulated object. In this article, we propose an approach to optimally control the deformation of linear and planar deformable objects while also minimizing the deformation energy of the object. First, we modify the framework initially developed for linear soft robot control in order to adapt it to deformable object robotic manipulation. To do so, we reformulate the problem as an optimization problem where the whole shape of the object is taken into account instead of solely focusing on the tip of the object's position and orientation. We then include an energy term in the cost function to find the solution that minimizes the potential elastic energy in the manipulated object while reaching the desired shape. Solutions to problems with high non-linearities are notoriously difficult to find and sensitive to local minima. We define intermediate optimal steps connecting the known initial and final configurations of the object and sequentially solve the problem, thus enhancing the robustness of the algorithm and ensuring the optimality of the solution. The intermediate optimal configurations are then used to define an end-effector trajectory for the robot to deform the object from an initial to the desired configuration.

Index Terms—Deformable object manipulation, Robotics, Shape control, Optimization, Trajectory generation

I. INTRODUCTION

Robotic manipulation of deformable objects has a wide range of applications, from food handling [1] to cable manipulation [2], [3] in order to automate the production of motors [4], [5], garment manipulation [6] or robotic surgery [7], [8]. Deformable objects are everywhere in our daily lives and the industry. However, the study of deformable objects has not received as much attention in the research community as rigid object manipulation. This is mainly due to the challenging problems manipulating these objects raise. Deformable objects have very high degrees of freedom which not only make the classic rigid object manipulation techniques impossible to apply [9] but also very hard to model accurately. Recent advances in computer graphics, machine learning, and hardware development have resulted in new modeling and simulation tools and techniques that allow overcoming some of the classical issues raised when manipulating these objects [10]. As a result, the demand for robotic solutions to manipulate deformable objects has emerged [11]. To manipulate these objects successfully, creating a connection between the actions performed by the robot and the behavior of the manipulated object is necessary. Among the many recent articles in the literature attempting to handle this difficult issue, two classes of methods emerge: data-driven and model-based approaches.

To address the high dimensionality of deformable objects and the modeling difficulties that derive from it, data-driven approaches have been used to solve deformable object manipulation tasks, especially garments and planar objects [6], [12]. These methods are usually formulated as a supervised learning problem where the robot must replicate a desired behavior. The training data typically originate from human demonstration [12]. These approaches have the advantage of not relying on a specific model but they are often hard to generalize from a small amount of training data [13], [14]. Other methods use data from robotic exploration, and although an exhaustive exploration is rarely used, reinforcement learning approaches are often initialized with human demonstration [15]. The improvement in simulators opened the possibility to learn politics in simulated environments [6], [16]. However, the policies learned in simulation often face a sim-to-real gap, especially for deformable objects which are difficult to represent accurately in a simulator.

Model-based methods rely on a model of the object to predict the deformations. There is a wide range of models with different properties. Less physically realistic models like mass-spring systems [17] and position-based dynamics [18] tend to be computationally effective but can sometimes lack precision, especially for larger deformations for mass-spring systems while position-based dynamics does not accurately model force effect. Although most models are determined beforehand, approaches relying on visual information to estimate a model online have also been proposed [19], [20], [21], [22].

Recently, physically realistic continuum models such as Finite elements [23] have been used to model deformable objects for robotic applications despite their high computational cost. In [24], the authors, based on the framework proposed in [25], [26] for soft robot real-time control, proposed an

inversion method of the finite element model of a deformable object. The inverse FEM model is then used to determine in real-time the controls to apply at the fingertips of a robotic hand to deform the object towards a desired shape [27]. With a similar approach, Cosserat theory has been used to successfully model, control [28], [29] and estimate [30] the state of linear deformable robots. The Cosserat theory provides one of the most accurate mechanical model while still being computationally efficient. The authors in [31] proposed a real-time closed-loop control strategy of a soft linear robot using the Cosserat rod model. It is especially suited to beam-like and linear objects which makes this model adequate for cables, ropes, and linear robot control [32]. The formulation of the problem of shape control is very similar to the underactuated soft robot control problem. There is an analogy between the actuators in soft robot control and contact points in deformable object manipulations [24].

In this article, we propose a method to generate optimal robot end-effector trajectories to solve constrained deformable object manipulation problems in industrial environments. We achieve this by adapting the framework originally developed for soft robot control in [28], [29], [31] to deformable linear object manipulation and by combining the Cosserat rod model used to represent the object with an optimization approach. We then define intermediate optimal steps in order to generate the optimal robot trajectory to solve the problem with respect to its specificities.

The paper is organized as follows. First, we introduce the general framework for the Cosserat rod model and how the constitutive equations are derived. Then we present our approach to adapt the problem from soft robot control to deformable object manipulation and how we formulate and solve the optimization problem, resulting in an end-effector trajectory to reach the desired deformed shape. Lastly, we illustrate this approach with a manipulation application where we control the shape of a linear clamped object with a robotic arm.

II. CLASSICAL STATIC COSSERAT ROD MODEL

To define the shape of a linear deformable object, we have to consider the rod kinematics that provides the position of the curve in $\mathbb{R}^3$ and its orientation. The rod is spatially discretized so that the model constitutive equations are derived from the static equilibrium at any point of the rod. These equations give a relation between the forces and moments applied on the rod and its position and orientation. Therefore, the position and orientation of every section of the rod are entirely defined by the external loads, initial conditions, and material properties. In the following section we will rely on the methods developed by Antman in [33] and later applied to soft robotic manipulators in [34], [35], [28], [29]. Although these works focus on soft robotic manipulators, the geometric framework and formalism used to define the shape of a linear object can be used for robotic manipulation.

A. Kinematics

We consider a linear Cosserat rod of length L defined by the position $p(s) \in \mathbb{R}^3$ and orientation $R(s) \in SO(3)$ of every point of the reference parameter s $\in [0, L]$ of the rod. Therefore, $p(s_1)$ gives the position in $\mathbb{R}^3$ of the point s_1 m from the origin of the undeformed rod while $R(s_1)$ is the orientation in the global frame attached to the point s_1. As presented in [29], we can use the homogeneous rigid-body transformation $g(s) \in SE(3)$ to describe the configuration of any point s of the rod:

$$g(s) = \begin{bmatrix} R(s) & p(s) \\ 0 & 1 \end{bmatrix}$$

$v(s)$ and $u(s)$ represent the local changes in translation and rotation of an infinitesimally small section of the Cosserat rod with respect to s. They indicate the evolution of $g(s)$ along s:

$$\begin{aligned} \dot{p}(s) &= R(s)v(s), \\ \dot{R}(s) &= R(s)\hat{u}(s) \end{aligned} \tag{1}$$

Here, the hat ^ represents the bijective mapping from a vector to a skew-symmetric matrix while the dot indicates the spatial derivative along s.

B. Static equilibrium

Consider an arbitrary segment of the rod from a point s_1 to a point s_2 with $s_1 < s_2$ and $(s_1, s_2) \in [0, L]^2$. The static equilibrium equation on this segment gives:

$$n(s_2) - n(s_1) + \int_{s_1}^{s_2} f(\sigma)d\sigma = 0, \tag{2}$$

$$m(s_2) + p(s_2) \times n(s_2) - m(s_1) - p(s_1) \times n(s_1) \\ + \int_{s_1}^{s_2} (p(\sigma) \times f(\sigma) + l(\sigma))d\sigma = 0, \tag{3}$$

Where $n(s_2)$, $m(s_2)$ are respectively the internal forces and moments applied by the $[s_1, s_2]$ segment to the next segment of the rod $(]s_2, L])$ while $n(s_1)$, $m(s_1)$ are respectively the internal forces and moments applied by this segment to the previous segment of the rod $(]0, s_1[)$. f and l are the force and moment distributions applied along the segment and the terms $\int_{s_1}^{s_2} f(\sigma)d\sigma$ and $\int_{s_1}^{s_2} (p(\sigma) \times f(\sigma) + l(\sigma))d\sigma$, therefore, represent the external loads applied to the whole segment as shown in Figure 1 for a section from 0 to L. By taking the partial derivative form with respect to s the equations 2 and 3 become the constitutive equations for internal force and moment:

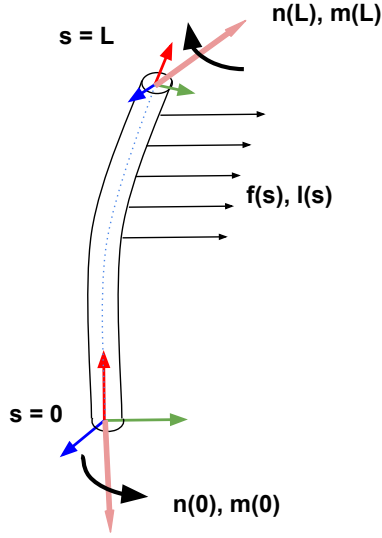$$\begin{aligned} \dot{n}(s) + f(s) &= 0, \\ \dot{m}(s) + \dot{p}(s) \times n(s) + l(s) &= 0 \end{aligned} \tag{4}$$

Fig. 1: Section from 0 to L of a Cosserat rod, deformed under the action of external forces f and moments l. Internal forces n and moments m are also represented at s = 0 and s = L.

C. Constitutive Equations

In this section, we use the formulation developed by Rucker in a series of works ([28], [29], [36]. The constitutive equations relate the deformation of the rod to the internal forces and moments and external loads applied on the rod. The rod is made of a linear elastic material that respects Hooke's law: $F = k\Delta x$. The notation $\sim$ refers to the undeformed configuration of the rod. Therefore $\tilde{p}(s)$, $\tilde{R}(s)$, $\tilde{v}(s)$ and $\tilde{u}(s)$ refer to the initial rest position, orientation and local changes in translation and rotation of the reference rod.

The stiffness matrices, for shear and extension K_{tra} and for bending and torsion K_{rot} are defined as:

$$K_{tra} = \begin{bmatrix} A(s)G & 0 & 0 \\ 0 & A(s)G & 0 \\ 0 & 0 & A(s)E \end{bmatrix}$$
$$K_{rot} = \begin{bmatrix} I_x(s)E & 0 & 0 \\ 0 & I_y(s)E & 0 \\ 0 & 0 & J(s)G \end{bmatrix} \tag{5}$$

Where I_x and I_y are the second moments of area of the rod cross-section, J is the polar moment of inertia of the cross-section ($J = I_x + I_y$). E and G are the Young's and shear moduli. From the linear constitutive laws are then derived the following relations between internal forces and moments and local changes $v(s)$ and $u(s)$, expressed in the local coordinate frame:

$$n(s) = R(s)K_{tra}(s)(v(s) - \tilde{v}(s)),$$
$$m(s) = R(s)K_{rot}(s)(u(s) - \tilde{u}(s)) \tag{6}$$

From 6 it is possible to obtain an expression of v and u with respect to s:

$$v(s) = K_{tra}^{-1}R^T(s)n(s) + \tilde{v}(s),$$
$$u(s) = K_{rot}^{-1}R^T(s)m(s) + \tilde{u}(s) \tag{7}$$

Equations 1, 4 and 8 then constitute the following system of differential equations:

$$\begin{aligned} \dot{p} &= Rv, & v &= K_{tra}^{-1}R^Tn + \tilde{v}, \\ \dot{R} &= R\hat{u}, & u &= K_{rot}^{-1}R^Tm + \tilde{u}, \\ \dot{n} &= -f, \\ \dot{m} &= -\dot{p} \times n - l \end{aligned} \tag{8}$$

III. MODEL-BASED OPTIMIZATION UNDER CONSTRAINTS

A. Shape Control Approach

In soft robot control applications [28], [36], [31], the emphasis is put on the actuation of the tip of the robot. In these works, the authors find the solution to the non-linear equations 8 that respect the desired position and orientation of the tip of the robot. While this approach is efficient when controlling the position of a single point and in general for soft robot actuation, it is not adapted for general linear object manipulation as it struggles to achieve complex object shapes. To accurately control the shape of a deformable object with a robotic arm, it is needed to solve an underactuated problem with limited grasping points to deform an object with a large number of degrees of freedom.

We consider the configuration of the rod defined by the configuration of every point of the rod (ie. position p, orientation O, internal forces n, and moments m of every point of the rod). In order to adapt the approach to deformable object manipulation, we define a low-level representation consisting of a set of k objective points carefully selected such that their position represents the desired shape of the manipulated object. See Figure 2 where the objective points (red) represent the desired shape. The objective is to deform the manipulated object in order to match the shape defined by the set of objective points. In [23], [24], the authors minimize the position (and orientation) error between a set of desired positions and a set of pre-defined control points on the object. This approach works best when the number of pre-defined points whose positions are controlled is equal to the number of actuated points, hence why it is used in soft robot actuation. However, for shape control applications it is needed to control multiple points with partially coupled positions. These coupling constraints considerably limit the set of feasible solutions even though a solution that produces the desired shape exists when picking different control points with different couplings. Figure 2 illustrates this issue. It is impossible for the control points (blue) to match the position of the objective points (red) because it would require the rod to both stretch and compress at different locations, which would require at least three grasping points, although a solution exists when selecting other control points.
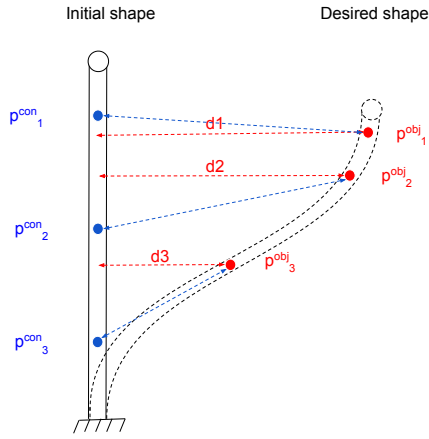
Fig. 2: Representation of the issue with the classic control points (blue) - objective points (red). We propose instead to minimize the distances d_1, d_2, d_3

We propose instead to consider the curve Γ formed by the linear object in the 3D space and to minimize the distance error between the set of k objective points and Γ. Thereby we do not control the position of a few control points and instead control the shape of the object so that the distance between Γ and the objective positions is minimal. The result is a curve Γ that passes through every desired point but without additional constraints on the positions of specific control points. The number of objective points k is defined arbitrarily but the higher k is, the more important the constraints on the desired configuration are and the harder it is to find a solution. We then approximate the target shape with B-splines to ensure its feasibility.

B. Deformation Energy

While this approach is more effective in finding solutions that respect a desired shape, it has the disadvantage of generating multiple solutions. To rank these solutions, we introduce the deformation energy E as a secondary objective. In [23], the authors compute the FEM energy to determine a stable robot pose. In our case, we search the solution that minimizes E, which ensures the internal strains in the object are minimal. This is especially useful for industrial applications where objects must be handled with care. We suppose the system is quasi-static and the gravity negligible:

$$
\begin{aligned}
E &= \int_0^L e(s)ds, \\
&= \int_0^L \frac{1}{2}(v(s) - \tilde{v}(s))^T(s)K_{tra}(s)(v(s) - \tilde{v}(s)) \\
&\quad + \frac{1}{2}(u(s) - \tilde{u}(s))^T K_{rot}(s)(u(s) - \tilde{u}(s))ds
\end{aligned}
\tag{9}
$$

Since integrating the Cosserat equations provides the full configuration of the rod, the only unknowns are the initial conditions that correspond to the configuration of the first point

of the rod. The solution are the initial conditions $p(0)$, $O(0)$ $n(0)$, $m(0)$ that lead to an optimal configuration of the rod. The boundary conditions depend on the type of grasping at the limits of the object, for realistic grasping situations, additional constraints must be set to reflect the grasping uncertainty (ie. $n_x = 0$ or $m_x = 0$ if the grasp does not enable forces or moments along the x axis). For a linear object of length L, clamped at one extremity and grasped by a robotic arm at the other, the objective function is defined as:

$$
\begin{aligned}
\min_{n(0),m(0)} F &= \frac{W_1}{2L} \int_0^L (v(s) - \tilde{v}(s))^T K_{tra}(s) \\
&(v(s) - \tilde{v}(s)) + (u(s) - \tilde{u}(s))^T K_{rot}(s) \\
&(u(s) - \tilde{u}(s))ds + W_2 \sum_{i=1}^k dist(\Gamma, p^{obj}(s_i)),
\end{aligned}
\tag{10}
$$

subject to:

$$
\begin{cases}
p(s = L) \subset Ws, \\
\dot{p} = Rv, \\
\dot{R} = R\hat{u}, \\
\dot{n} = -f, \\
\dot{m} = -\dot{p} \times n - l
\end{cases}
$$

We have a weighted (W_1, W_2) objective function where $p^{obj}(s_{i=1..k})$ are the objective points positions and Ws is the operational workspace of the robot's end-effector (see Figure 3).
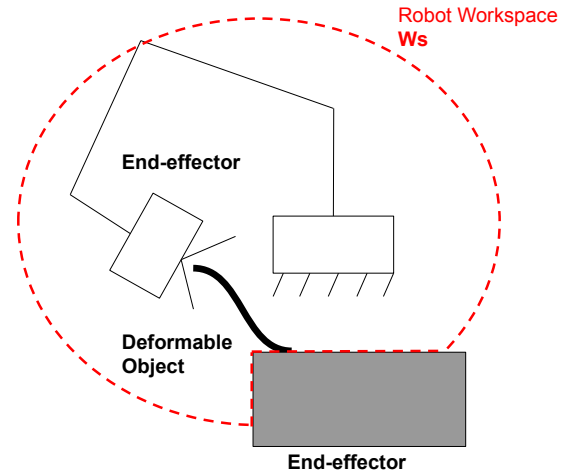

Fig. 3: schematic representation of a robotic end-effector's operational workspace Ws

C. Numerical resolution

The Cosserat forward kinematic model is useful to predict the deformations of the object but for a motion planning application, we need to be able to generate the command to send to the robot in order to control the shape of the manipulated object. However, the Cosserat forward kinematic
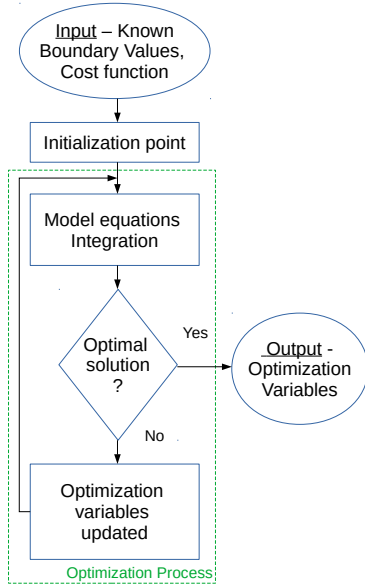
Fig. 4: Optimization process

Fig. 5: Set of n_{inter} successive optimal configurations from which the end-effector trajectory is derived

model cannot be easily inverted so we cannot use an analytical solution. Instead, we numerically solve the problem with an algorithm based on the shooting method which is a classical approach for solving boundary value problems. The Cosserat equations are integrated along the rod until boundary conditions that satisfy the constraints are found. On top of this, there is another optimization loop that ensures the optimality of the solution where the optimization variables are the unset initial conditions of the problem. When integrating the model equation with the optimized variables, we obtain an optimal configuration of the rod with respect to the cost function (see Figure 4).

D. Optimal trajectory generation

Interior-point methods are used for the optimization therefore we need to provide the algorithm an initial guess "close enough" to the solution to start the optimization. Since this problem is a highly nonlinear boundary value problem, providing a good initial guess is a key problem as the algorithm can fail to avoid local minima or to find a solution otherwise.

To ensure the algorithm does not fall in local minima, we introduce n_{inter} intermediate object configurations between the initial known and final configurations. To do this, we define p^{init} as the projection of p^{obj} on the initial shape and discretize the positions between p^{obj} and p^{init} into n_{inter} sets of intermediate objective points configurations. We then iterate and solve the optimization problem at each step by providing the results from the previous step as the initial guess for the next step.

As a by-product, we also obtain a set of intermediate optimal configurations that serve as a trajectory for the object to deform from its initial to the desired configuration. In [31], the trajectory is defined by discretizing the end-effector's position
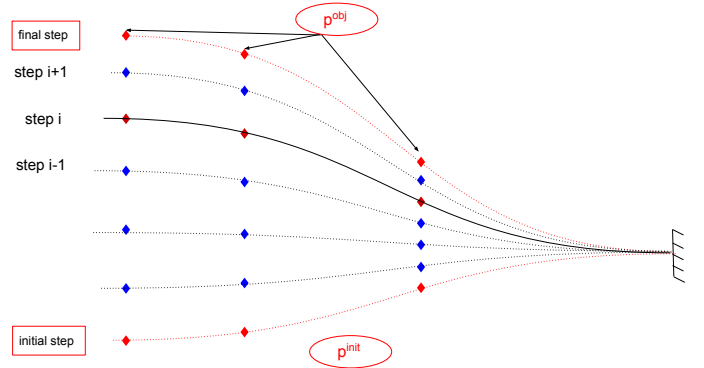
between the initial and final configurations but the state of the object is not known during the trajectory, the feasibility of the path is therefore not guaranteed. Another advantage is that for real applications the viscosity of the object is often not negligible, each configuration hence depends on the path taken and there is benefit in defining the intermediate steps and controlling the shape of the object during the manipulation. The intermediate set of objective positions is defined as:

$$p^i = p^{init} + (i-1)\frac{p^{obj} - p^{init}}{n_{inter}} \qquad (11)$$

Where $i = 1...n_{inter}$, see Figure 5 for an illustration.

We suppose that the contact between the robot and the object remains steady during the manipulation, therefore the configuration of the end-effector corresponds to the configuration of the contact point. From the output of the algorithm, we can subsequently deduce the position and orientation of the end effector to deform the object into this optimal configuration.

Since we use a static Cosserat model, the end-effector velocity has to be low enough to limit the dynamic effects.

The process can be summarized as (see Figure 6):

- Define p^{obj};
- Input p^{obj} and p^{init};
- Define the intermediate objectives between p^{obj} and p^{init};
- Solve the optimization process at each step, providing the input for the next step;
- Determine the end-effector commands from the intermediate and final configurations of the object.

IV. EXPERIMENTAL VALIDATION

1) Experimental Setup: To validate the described approach, we propose the following experiment: a robotic arm deforms a linear deformable object, clamped at one end and grasped by the robot at the other end, towards a desired 3D shape. We compare each object pose during the trajectory to the model and then also compare it to a reference trajectory. Let us consider the rubber band (see table I) clamped so that the initial position $p(0)$ and orientation $O(0)$ are fixed. We
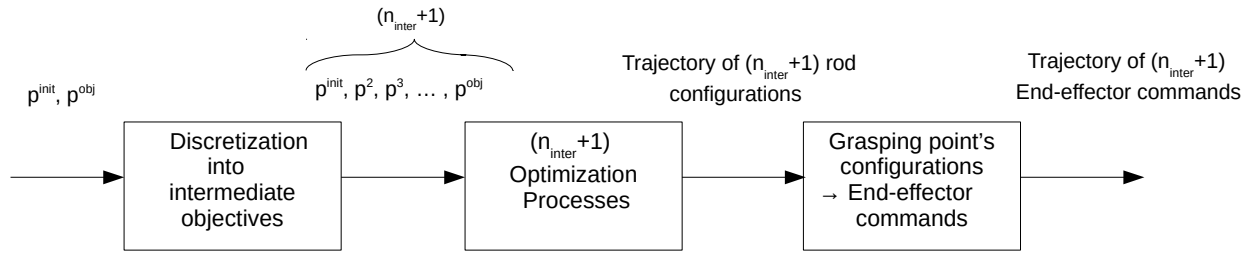
Fig. 6: Successive steps to acquire a trajectory from known initial and desired configurations
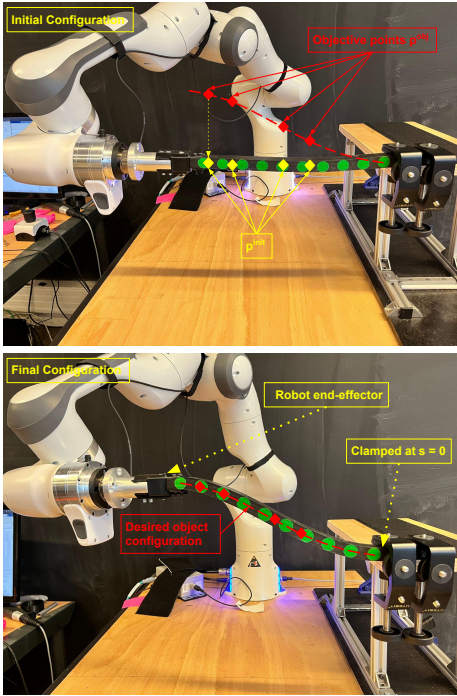

Fig. 7: Experimental setup at the initial and final configurations.

define $p^{obj} = \begin{bmatrix} p_1^{obj} p_2^{obj} p_3^{obj}...p_n^{obj} \end{bmatrix}^T$ the objective points that represent the desired shape in $\mathbb{R}^3$. The trajectory is generated by discretizing the position of the objective points between the desired and initial configurations into $n_{inter} = 7$ intermediate steps.

Parameter	Value
E	$3.2e^6 \ N/m^2$
ν	0.5
$G = \frac{E}{2(1+\nu)}$	$1.06e^6 \ N/m^2$
EI_x	$0.04 \ N.m^2$
EI_y	$0.04 \ N.m^2$
GJ	$0.028 \ N.m^2$
ρ	$1392 \ kg/m^3$
L	$0.29 \ m$
h	$0.02 \ m$
b	$0.02 \ m$
g	$9.81 \ m/s^2$

TABLE I: Parameters

For the experimental setup, we use a Franka Emika Panda

seven degrees of freedom arm (see Figure 7). The robot is controlled with Moveit and ROS using Franka Emika FCI, developed for research applications. The rubber band (see table I) is clamped at one end and fixed to the robot thanks to a 3D printed handle that maintains steady the rubber band so that the orientation and position of the end effector correspond to the position and orientation of the extremity of the rubber band (see Figure 7). The setup also includes a 3D Kinect V2 that tracks the position of 10 markers evenly distributed on the rubber band.

2) Results: To validate the approach, we compare the object's configuration to p^{obj} and plot the error over time to show the convergence of the algorithm. The error is the average distance between the measured configuration of the object and each objective point of p^{obj}. The robot's end-effector velocity is defined beforehand and is not a source of error as long as the model hypotheses are fulfilled, ie. the object's dynamic effects are negligible. To highlight the efficiency of the method over different objects and shapes, we propose five experiments each with different p^{obj} or objects and plot the error over time during the manipulation in Figure 9. The corresponding experiments are depicted in Figure 8, the upper picture is a photo of the system once the objective is reached and the lower graph represents a comparison between the objective points p^{obj}, the model's shape prediction, and the measured state of the object at the final state of the trajectory for the corresponding experiment. The final step 3D error distribution for the experiments is shown in Figure 10 and ranges from 1.6mm to 3.6mm. At this scale, the camera error is not negligible, especially for the depth measurements that are less precise with the Kinect V2 than in the other directions. We therefore also show the errors in the 2D plane in Figure 10 and 9b. Here we used two different objects, a beam of square section and an elastic band. Any object can be used as long as the Cosserat model's hypotheses are fulfilled. Since the velocity is constant across all experiments, the robot's travel time is only determined by the path taken.

To show that our approach is general and that the final error is stable over multiple $p^{obj]}$, we compare the error distribution obtained in Figure 10a to the error distribution obtained when repeating the same experiment 8 times with the same inputs (see Figure 10b). Repeatability is never ensured when manipulating deformable objects, multiple hypotheses like the elastic behavior and the negligible viscosity of the
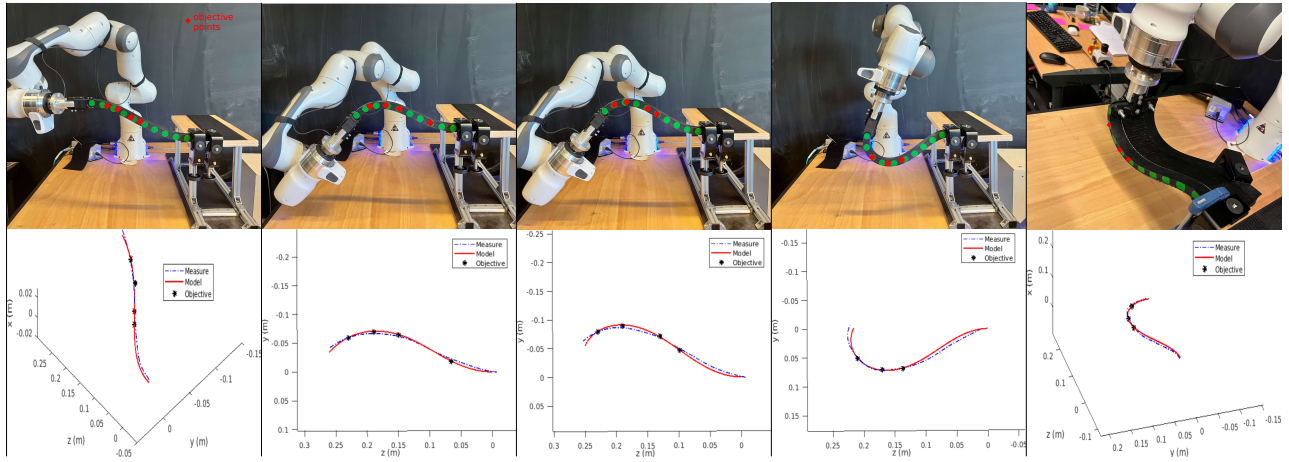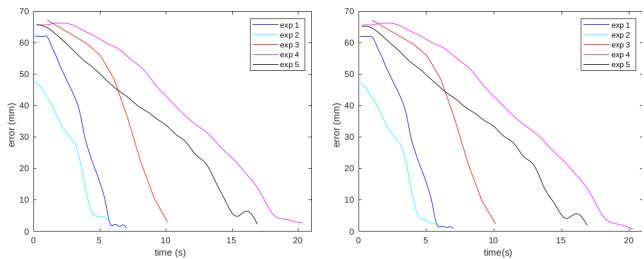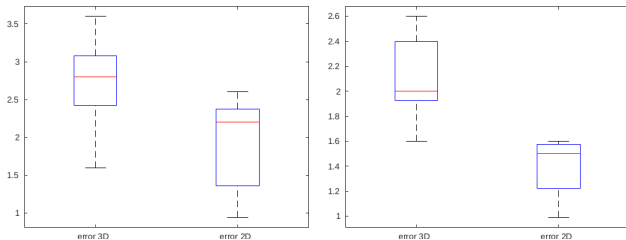
Fig. 8: Experimental results for multiple objectives and object shapes

(a) 3D Error convergence over time for the five experiments

(b) 2D Error convergence over time for the five experiments

Fig. 9

(a) Final error variance across multiple experiments with different p^{obj}

(b) Final error variance across the same experiment repeated multiple times

Fig. 10

object are not verified in reality. Hence, the shape of the object depends on the path taken unless the manipulation speed is slow enough to entirely nullify the viscous effects, which is not suited for robotic applications. Also, flaws in the object and errors in the estimation of elastic parameters often lead to unexpected behaviors and biases. These factors can hinder the repeatability of the experiment hence why we repeated the same experiment multiple times to estimate the error variance. We can see that the distribution for the repeated experiment lies in a similar range as the one with different experiments, which shows the consistency as well as the repeatability to some degree of the method. This must however be confirmed

with further experimentation as the sample size is low for both cases.

V. DISCUSSION AND FUTURE WORK

In this article, we proposed an approach based on the Cosserat theory to control the shape of linear and planar objects that respect the Cosserat hypothesis. We used the framework previously developed for deformable linear robot control and adapted it to deformable object manipulation. We proposed to reformulate the problem as an optimization problem where the distance between a set of objective points and the 3D curve formed by the object and the elastic energy in the object are minimized. To improve the robustness of the algorithm and control the trajectory connecting the initial and final configurations of the object, we discretized the objective points into intermediate objectives in order to define intermediate configurations. From these previous configurations, we deducted the end-effector positions to deform the object toward the desired shape. To further improve the results and the final distance error, the next step is to close the loop and add vision and force feedback. There is a trade-off between execution time and algorithm precision and multiple parameters can be influenced in order to reduce it, among them are the number of objective points, the complexity of the desired shape, the number of intermediate objectives, object rigidity, and discretization fineness, weights in the cost function. So far these parameters are tuned towards high precision but adding sensor feedback can help counterbalance the precision loss. Additionally, a spline or curvature-based representation of the object can be used to estimate the load [37] at the limits of the objects.

Two main challenges in this field are handling contacts with the manipulated object as well as manipulating an object with a complex shape. If the contact forces are known (ie. a clip on the object of a known punctual force), then both problems can be solved by having multiple integration sections with different boundary conditions and shapes. The problem is much more

difficult however for contacts that are difficult to model (ie. velcro, contact with other soft objects).

FUNDING SOURCES

REFERENCES

[1] Z. Wang and S. Hirai, "Finite element modeling and physical property estimation of rheological food objects," *Journal of Food Research*, vol. 1, no. 1, p. 48, 2012.

[2] 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)*. IEEE, 2018, pp. 479–484.

[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] O. Roussel and M. Taïx, "Deformable linear object manipulation planning with contacts," in *Robot Manipulation: What has been achieved and what remains to be done? Full day workshop at IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2014.

[5] A. J. Shah and J. A. Shah, "Towards manipulation planning for multiple interlinked deformable linear objects," in *2016 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2016, pp. 3908–3915.

[6] J. Matas, S. James, and A. J. Davison, "Sim-to-real reinforcement learning for deformable object manipulation," in *Conference on Robot Learning*. PMLR, 2018, pp. 734–743.

[7] S. Patil and R. Alterovitz, "Toward automated tissue retraction in robot-assisted surgery," in *2010 IEEE International Conference on Robotics and Automation*. IEEE, 2010, pp. 2088–2094.

[8] I. Leizea, A. Mendizabal, H. Alvarez, I. Aguinaga, D. Borro, and E. Sanchez, "Real-time visual tracking of deformable objects in robot-assisted surgery," *IEEE computer graphics and applications*, vol. 37, no. 1, pp. 56–68, 2015.

[9] Y.-B. Jia, H. Lin, and F. Guo, "Optimal two-finger squeezing of deformable objects," in *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2013, pp. 3514–3519.

[10] 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.

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

[12] J. Schulman, A. Gupta, S. Venkatesan, M. Tayson-Frederick, and P. Abbeel, "A case study of trajectory transfer through non-rigid registration for a simplified suturing scenario," in *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2013, pp. 4111–4117.

[13] E. Pignat and S. Calinon, "Learning adaptive dressing assistance from human demonstration," *Robotics and Autonomous Systems*, vol. 93, pp. 61–75, 2017.

[14] J. Schulman, J. Ho, C. Lee, and P. Abbeel, "Learning from demonstrations through the use of non-rigid registration," in *Robotics Research*. Springer, 2016, pp. 339–354.

[15] Y. Tsurumine, Y. Cui, E. Uchibe, and T. Matsubara, "Deep reinforcement learning with smooth policy update: Application to robotic cloth manipulation," *Robotics and Autonomous Systems*, vol. 112, pp. 72–83, 2019.

[16] A. Clegg, W. Yu, J. Tan, C. K. Liu, and G. Turk, "Learning to dress: Synthesizing human dressing motion via deep reinforcement learning," *ACM Transactions on Graphics (TOG)*, vol. 37, no. 6, pp. 1–10, 2018.

[17] S. Hirai, T. Tsuboi, and T. Wada, "Robust grasping manipulation of deformable objects," in *Proceedings of the 2001 IEEE International Symposium on Assembly and Task Planning (ISATP2001). Assembly and Disassembly in the Twenty-first Century.(Cat. No. 01TH8560)*. IEEE, 2001, pp. 411–416.

[18] J. Bender, M. Müller, and M. Macklin, "Position-based simulation methods in computer graphics." in *Eurographics (tutorials)*, 2015, p. 8.

[19] D. Navarro-Alarcon, Y.-H. Liu, J. G. Romero, and P. Li, "Model-free visually servoed deformation control of elastic objects by robot manipulators," *IEEE Transactions on Robotics*, vol. 29, no. 6, pp. 1457–1468, 2013.

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

[21] 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*, vol. 32, no. 2, pp. 429–441, 2016.

[22] 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*, vol. 34, no. 1, pp. 272–279, 2017.

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

[24] 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)*. IEEE, 2018, pp. 4007–4013.

[25] E. Coevoet, T. Morales-Bieze, F. Largilliere, Z. Zhang, M. Thieffry, M. Sanz-Lopez, B. Carrez, D. Marchal, O. Goury, J. Dequidt *et al.*, "Software toolkit for modeling, simulation, and control of soft robots," *Advanced Robotics*, vol. 31, no. 22, pp. 1208–1224, 2017.

[26] C. Duriez, E. Coevoet, F. Largilliere, T. Morales-Bieze, Z. Zhang, M. Sanz-Lopez, B. Carrez, D. Marchal, O. Goury, and J. Dequidt, "Framework for online simulation of soft robots with optimization-based inverse model," in *2016 IEEE International Conference on Simulation, Modeling, and Programming for Autonomous Robots (SIMPAR)*. IEEE, 2016, pp. 111–118.

[27] F. Faure, C. Duriez, H. Delingette, J. Allard, B. Gilles, S. Marchesseau, H. Talbot, H. Courtecuisse, G. Bousquet, I. Peterlik *et al.*, "Sofa: A multi-model framework for interactive physical simulation," in *Soft tissue biomechanical modeling for computer assisted surgery*. Springer, 2012, pp. 283–321.

[28] D. C. Rucker, B. A. Jones, and R. J. Webster III, "A geometrically exact model for externally loaded concentric-tube continuum robots," *IEEE transactions on robotics*, vol. 26, no. 5, pp. 769–780, 2010.

[29] D. C. Rucker and R. J. Webster III, "Statics and dynamics of continuum robots with general tendon routing and external loading," *IEEE Transactions on Robotics*, vol. 27, no. 6, pp. 1033–1044, 2011.

[30] S. Lilge, T. D. Barfoot, and J. Burgner-Kahrs, "Continuum robot state estimation using gaussian process regression on se (3)," *The International Journal of Robotics Research*, p. 02783649221128843, 2022.

[31] F. Campisano, S. Caló, A. A. Remirez, J. H. Chandler, K. L. Obstein, R. J. Webster III, and P. Valdastri, "Closed-loop control of soft continuum manipulators under tip follower actuation," *The International journal of robotics research*, vol. 40, no. 6-7, pp. 923–938, 2021.

[32] D. K. Pai, "Strands: Interactive simulation of thin solids using cosserat models," in *Computer graphics forum*, vol. 21, no. 3. Wiley Online Library, 2002, pp. 347–352.

[33] J. H. P. H. J. Keener and J. K. B. M. A. Mielke, "Ss antman je marsden l. sirovich," 2005.

[34] D. Trivedi, A. Lotfi, and C. D. Rahn, "Geometrically exact models for soft robotic manipulators," *IEEE Transactions on Robotics*, vol. 24, no. 4, pp. 773–780, 2008.

[35] B. A. Jones, R. L. Gray, and K. Turlapati, "Three dimensional statics for continuum robotics," in *2009 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2009, pp. 2659–2664.

[36] D. C. Rucker and R. J. Webster, "Computing jacobians and compliance matrices for externally loaded continuum robots," in *2011 IEEE International Conference on Robotics and Automation*. IEEE, 2011, pp. 945–950.

[37] M. A. Diezinger, B. Tamadazte, and G. J. Laurent, "3d curvature-based tip load estimation for continuum robots," *IEEE Robotics and Automation Letters*, vol. 7, no. 4, pp. 10 526–10 533, 2022.