Closed-Loop Shape Control of Deformable Linear Objects Based on Cosserat Model

Azad Artinian, Faïz Ben Amar, Véronique Perdereau

Closed-loop shape control of deformable linear objects based on Cosserat model

Azad Artinian[1], Faïz Ben Amar[1], and Véronique Perdereau[1]

***Abstract*—The robotic shape control of deformable linear objects has garnered increasing interest within the robotics community. Despite recent progress, the majority of shape control approaches can be classified into two main groups: open-loop control, which relies on physically realistic models to represent the object, and closed-loop control, which employs less precise models alongside visual data to compute commands. In this work, we present a novel 3D shape control approach that includes the physically realistic Cosserat model into a closed-loop control framework, using vision feedback to rectify errors in real-time. This approach capitalizes on the advantages of both groups: the realism and precision provided by physics-based models, and the rapid computation, therefore enabling real-time correction of model errors, and robustness to elastic parameter estimation inherent in vision-based approaches. This is achieved by computing a deformation Jacobian derived from both the Cosserat model and visual data. To demonstrate the effectiveness of the method, we conduct a series of shape control experiments where robots are tasked with deforming linear objects towards a desired shape.**

***Index Terms*—Dual Arm Manipulation, Materials Handling, Modeling, Control, and Learning for Soft Robots, Sensor-based Control.**

I. Introduction

Robotic manipulation of deformable objects, and specifically the shape control of deformable objects is a growing subject in the research community. Its potential applications are as diverse as common since deformable objects are everywhere, from food manipulation [1] to robotic surgery [2] and garments folding [3]. One particularly important object type is Deformable Linear Objects (DLO), such as cables and beams, which are prevalent across various manufacturing sectors and have received significant attention from the community in recent years [4], [5].

Multiple methods in the literature have been used to solve shape control tasks, either for deformable robots or objects. Mechanical models such as finite elements method (FEM) [6], [4] or Cosserat-based models [7] are powerful tools to achieve such goals. They allow the creation of a direct relationship between the inputs of the robots and the resulting object shape. However, despite being among the most precise models available, they are highly sensitive to the estimation of elastic parameters and are computationally expensive. The stiffness matrices, which are derived from these elastic parameters, require precise estimation, which can be difficult. Additionally, due to the high computational costs, most controllers using these models are open-loop [6], [8]. The authors in [7] achieved a closed-loop approach to control a deformable linear robot (DLR), but DLR control focuses more on the displacement of the tip than on the full shape of the robot and this approach can hardly be transposed to object manipulation. In [4], the authors proposed a closed-loop controller coupled with a reduced FEM model to control a clamped DLO.

To avoid the high computational costs and achieve closed-loop control, other approaches have focused on geometrical models or model-free methods. Usually, a local deformation Jacobian is computed to create a connection between the robot's action and the object's shape. In a series of work, the authors in [9], [10], in [11] and [5] applied visual-servoing approaches to robotic deformable object manipulation. The local deformation Jacobian matrix is initialized and then updated at each iteration. This Jacobian matrix represents the connection between the command sent to the actuators and a low-level representation of the shape of the manipulated object. This representation can either be the 2D contour [10], [11] or the positions of a set of control points on the object [12], [13]. The Jacobian matrix is updated based on the comparison between the visual data and the commands sent on a previous time window. It therefore depends on the anterior motions of the robot. Potential motions can therefore remain unexplored or be outdated if these degrees of freedom have not been used recently. This is especially problematic for more rigid deformable objects whose dynamics depend heavily on their current pose. To ensure every possible motion is taken into account when computing a solution, the authors in [14] introduced an ARAP model in the controller. The visual data are used to update the model which, in turn, is used to predict the next potential motions and compute the Jacobian. This approach has then been extended with an optimal controller in [15]. However, non-physics-based geometrical models can provide unfeasible or unrealistic shapes and can sometimes lack precision, especially for large deformations where more complex modes of deformation (Legendre polynomials) are difficult to capture with such models.

In this article, we propose a novel closed-loop approach to control the shape of DLOs. The goal is to preserve the realism of mechanical models while having none of their usual drawbacks. Our approach, based on Cosserat theory, provides an efficient and precise shape controller for DLOs, even for tasks involving complex desired shapes and deformations. The main advantages of our approach are its accuracy and

Manuscript received: February, 22, 2024; Revised July, 19, 2024; Accepted August, 14, 2024.

This work was supported by the SOFTMANBOT project, which received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 869855.

[1]Authors are with the Institute of Intelligent Systems and Robotics (ISIR), Sorbonne University artinian@isir.upmc.fr

Digital Object Identifier (DOI): 10.1109/LRA.2024.3451368

repeatability, without requiring prior precise knowledge of the manipulated object, as well as its generality since it handles complex objects made with non-isotropic, non-homogeneous composite materials. Additionally, by integrating the Cosserat equations at every iteration, we ensure that the generated shape is feasible and physically realistic.

The paper is organized as follows. First, we present the Cosserat framework from which are derived the constitutive ordinary differential equations (ODE) that govern the shape of the object. Then, we propose to formulate the shape control problem as a global boundary value problem (BVP) where the manipulated object is held at both limits by the robot. We then include the Cosserat ODE in our control scheme that aims at solving an initial value problem (IVP) at each iteration to solve the BVP globally. Finally, we test our approach on our experimental setup consisting of two robotic arms and several objects with different properties.

II. Cosserat rod model

The Cosserat rod model is one of the most accurate mechanical models for DLOs. It has been used extensively in the soft robotics community to model [16], [17], control [18], [19], [7] and estimate [20] the shape and the dynamics of DLRs. In this work, we will apply the framework introduced by the authors in [21] and developed in [22], [18], [19]. It has the advantage of formulating the problem as an initial value problem.

As seen in chapter eight of [21], the Cosserat constitutive equations can be derived by applying static equilibrium on a section of the DLO:

$$\begin{aligned} \dot{p} &= Rv & v &= K_t^{-1}R^T n + \tilde{v} \\ \dot{R} &= R\hat{u} & u &= K_r^{-1}R^T m + \tilde{u} \\ \dot{n} &= -f \\ \dot{m} &= -\dot{p} \times n - l \end{aligned} \tag{1}$$

The Cosserat ODE govern the state $\Gamma = [p, R, n, m]^T$ of the object i.e. the position p, orientation R, internal forces n, and moments m at every point of the object. f and p are the external forces and moments, K_t and K_r the translational and rotational stiffness matrices, v and u the linear and angular rates of change, $\hat{}$ and $\sim$ refer respectively to the bijective mapping from $\mathbb{R}^3$ to $\mathfrak{so}(3)$ and the undeformed rest configuration of the object.

We consider the reference length parameter $s \in [0, L]$ for a rod of length L. If we know the vector of initial values $\Gamma_0 = \Gamma(s = 0) = [p_0, R_0, n_0, m_0]^T$, we solve the IVP by spatially integrating equations (1) with respect to s, and thus obtain the full state Γ at every point of the rod.

In other words, the full state Γ of the object can be very easily deduced if we know the initial values Γ_0 at $s = 0$. The forward Cosserat model is well-posed [19], we have 12 equations (see equation 1) as well as 12 unknowns in Γ at every point of the rod. Therefore, the solution obtained by solving the IVP is unique.

III. Problem Formulation

We use the classical problem formulation based on control and objective points. This formulation has been used extensively in the literature [6], [14], [4]. It has the advantage of being fast to compute and easy to set up.

We define two sets of points:

- The first set of objective points which position p^i_{obj}, with $i \in [1, n_c]$, represent the target shape of the object.
- The second set of control points, selected on the object, which positions p^i_c, with $i \in [1, n_c]$, are controlled.

Each objective point is associated with a single control point and the goal of the algorithm is to minimize the position error between each objective-control point couple: $\epsilon_p = \begin{bmatrix} \|p^1_{obj} - p^1_c\|_2 & \dots & \|p^n_{obj} - p^n_c\|_2 \end{bmatrix}^T = \begin{bmatrix} \epsilon^1_p & \dots & \epsilon^n_p \end{bmatrix}^T$. Figure 1 illustrates a schematic representation of the problem.

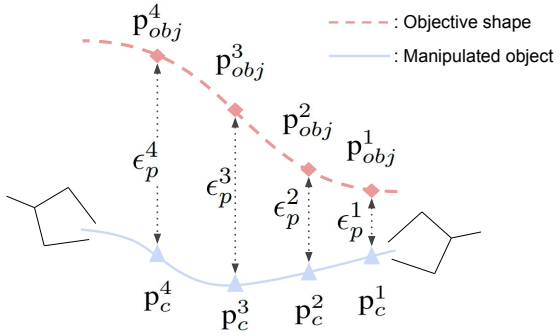


Fig. 1: Problem formulation: minimize ϵ_p

We then include the Cosserat equations that model the object to define the following BVP: find the initial values Γ_0 that, when integrated, provide a state of the object Γ that minimizes ϵ_p.

While solving the IVP is fast and efficient and only requires a spatial integration if we know the type of boundary conditions (i.e. clamped, free at the limits, etc.), solving the BVP, i.e. finding the initial values Γ_0 that respect the boundary conditions is much more difficult. Previous approaches have relied on optimization methods to find a numerical solution [19], [17], [23], but the computational cost of these algorithms makes them difficult to include in a closed-loop control scheme.

Our proposed approach is designed to preserve the precision of the Cosserat model while also making it less computationally expensive to include it in our closed-loop control scheme. Instead of directly solving the costly BVP, we propose to start from a known initial state that corresponds to the shape of the manipulated object at $t = 0$. At each iteration we estimate, using sensor data, the initial values Γ_0 that correspond to the current shape of the object. Then, we compute a local Jacobian matrix from both the sensor data and the Cosserat equations. We use this Jacobian to update Γ_0 and solve the new IVP to obtain the updated state Γ. From this updated state, we deduce the command to send to the robots to update the shape of the actual object. The successive computed initial values Γ_0 converge toward the global solution of the BVP, and therefore the shape of the manipulated object converges toward the desired shape.

IV. CLOSED-LOOP SCHEME

A. Deformation Jacobian based on Cosserat Model

The deformation Jacobian matrix J_d is a linear approximation of the Cosserat model locally. In most applications [7], [14], the authors compute the Jacobian by introducing a simulated disturbance in the input of the robots and observing the resulting simulated position of the control points. Instead, we introduce a disturbance directly in the initial configurations Γ_0 of the Cosserat IVP representing the object in its current shape. This allows us to update, at each iteration, Γ_0 and then solve the Cosserat IVP, thus ensuring the feasibility and physical realism of the resulting shape.

J_d creates a coupling between the variations of Γ_0 and the variations of the positions of the control points p_c^i. To ensure the local approximation is correct, we compute a new Jacobian at every time step instead of updating an existing one.

Computing J_d only requires to know the initial values Γ_0 corresponding to the current state of the object. For each component of Γ_0, we successively introduce a disturbance Δ_d^j with $j \in [1, m_l]$ and m_l the number of components of Γ_0. We then integrate the Cosserat equations from the disturbed initial conditions. Once we obtain the disturbed shape of the object, we compare the disturbed and current position of the control points and compute, with finite differences the column of J_d corresponding to the disturbed component of Γ_0. By repeating this process m_l times, we then obtain the full deformation Jacobian J_d which provides a coupling between the variation of each dimension of Γ_0 and their effect on the position of the control point p_c^i.

The type of robotic grip on the object directly affects the Jacobian matrices and the type of boundary conditions. For a DLO grasped and controlled at both ends, we have $m_l = 12$, and the size of Γ_0 is necessarily 12, 3 for the position, 3 for the orientation, and 6 for the internal forces and moments.

However for more realistic cases, it is often difficult to respect the clamped boundary conditions at the grasping points between the robot and the object, there can be for example friction or slippage along certain DOFs. Therefore, $m_l < 12$ and the perturbations Δ_d^j must be set to zero for those uncontrollable DOFs which produces a column of zeros in J_d^i. Following the Cosserat theory, if the gripper cannot apply forces and moments along specific directions at the limits, the internal forces and moments are constant along these directions all across the rod.

For a real-world application, the vector of initial values Γ_0 that corresponds to the current shape of the object is unknown. Therefore we have to estimate it at each iteration from the sensor data. In the following, we call this estimation $\hat{\Gamma}_0$.

B. Control law

Once J_d is obtained, we use it to compute the new initial values $\Gamma_{0,k+1}$ that will be used to update the shape of the object. We directly use the pseudo-inverse of J_d as well as the error ϵ_p at step k to compute the new initial values $\Gamma_{0,k+1}$ at step $k+1$:

$$\Gamma_{0,k+1} = \hat{\Gamma}_{0,k} + K J_d^{\dagger} \epsilon_p \tag{2}$$

where $\hat{\Gamma}_{0,k}$ is the estimation of the initial values vector, K is a parameter that affects the convergence rate toward the objective as well as the precision of the algorithm and $J_d^{\dagger}$ is the Moore-Penrose inverse of the Jacobian. The smaller K is the smaller the steps between each new Γ_0 are. The algorithm will therefore require more iterations to converge toward the global solution but will also be more precise. Since the Jacobian is a local approximation of the Cosserat model, it is therefore more accurate when computing solutions close to the Γ_0 where J_d was computed.

Once the desired initial conditions $\Gamma_{0,k+1}$ are computed, we solve the initial value problem to acquire Γ_{k+1}. From this full state, we then deduce the desired robotic end-effector position and orientation, and then derive a proportional control law to control the end effector's velocity and angular velocity and drive it toward the desired position and orientation, see equation 3:

$$\begin{aligned} v_t &= -K_t e_t \\ v_a &= -K_a e_a \end{aligned} \tag{3}$$

where v_t and v_a are the end-effector's velocity and angular velocity, K_t and K_a the associated gains, and e_t and e_a the position and orientation errors between the current end effector's pose and the new pose deduced from Γ_{k+1}. We use the same control law for both arms.

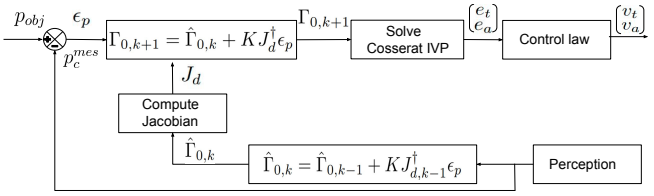


Fig. 2: Bloc diagram representing the control loop

C. Perception feedback

To compute J_d, we need to know the initial values $\Gamma_0 = [p_0, R_0, n_0, m_0]^T$ that correspond to the current state of the manipulated object. It is important to have an accurate estimation of Γ_0 since it is the reference point from which J_d is computed. While the initial position p_0 and orientation R_0 are easily deduced from either the robot at the gripping point or the type of boundary conditions, the main challenge is to accurately estimate the initial internal forces n_0 and moments m_0. Two approaches could be used to estimate Γ_0, one based on force sensors and the other one based on vision:

- Force sensor based: The most straightforward approach. If the robot is equipped at its tip with force sensors we can simply read the force and moment values at the limits and subtract external forces and moments to obtain n_0 and m_0. The main drawback of this approach, apart from requiring two force-torque sensors, is that these measurements need to be filtered to suppress noise. This adds a delay in the loop as small noises can be amplified and generate non-negligible errors. p_0 and R_0 are then directly obtained from the robot's end-effector feedback to obtain Γ_0.

- Vision based: This approach requires more steps but is more robust to noise or if feedback from the robot is not available. From the current error measurements of the control points ϵ_p, we use the initial conditions and the Jacobian computed at the previous step to estimate Γ_0 at the current step, see equation 4.

$$\hat{\Gamma}_{0,k} = \hat{\Gamma}_{0,k-1} + K J^{\dagger}_{d,k-1} \epsilon_p \tag{4}$$

While the visual approach is both much easier to set up and more robust to noise, it requires precise estimation of Γ_0 at the first step as well as longer computation time and is also sensible to occlusions. We represent the control loop on the figure 2 and on the algorithm 1.

Algorithm 1 Control loop at step t

$J_{d,k-1}$ // Jacobian computed at step $k-1$
$n_{0,k-1}$ // internal forces computed at step $k-1$
$m_{0,k-1}$ // internal moments computed at step $k-1$

while 1=1 **do**
$p_c^{mes} = getfromvision()$
$\epsilon_p = p_{obj} - p_c^{mes}$

$\hat{\Gamma}_{0,k} = \hat{\Gamma}_{0,k-1} + K J^{\dagger}_{d,k-1} \epsilon_p$ // **estimate current initial conditions**

$J_{d,k} = ComputeJacobian(\Gamma_{0,k})$
$\Gamma_{0,k+1} = \hat{\Gamma}_{0,k} + K J^{\dagger}_{d,k} \epsilon_p$
$[e_t, e_a]^T = SolveCosseratIVP(\Gamma_{0,k+1})$ // **EE position and orientation error**
$[v_t, v_a] = ControlLaw(e_t, e_a)$ // **EE velocities**
end while

D. Well-posed problem and stability

The Cosserat IVP is well-posed [21]. For each vector of initial values Γ_0, there exists a unique solution Γ obtained by integrating the Cosserat equations (equation 1) from Γ_0. Furthermore, the solution Γ changes continuously with Γ_0. Thus, the well-posedness of the Cosserat problem guarantees a unique solution as long as the material's elastic behavior assumptions are satisfied.

Furthermore, if Γ_0 is updated continuously during the manipulation, then the shape of the object also varies continuously. The stability conditions can therefore be reduced to the classic visual-servoing conditions discussed in [24], i.e. we can show local stability around the equilibrium points. We define our stability conditions in a similar way as presented by the authors in [25] for deformable robot control, more recently used in [7]. From J_d in $s = 0$, we extract the submatrix related to the variation of internal forces and moments and define our effective rigidity matrix:

$$M = J_d^{\dagger}(:, 6:12) \tag{5}$$

Contrary to the previous works on deformable robots, we define the rigidity matrix at $s = 0$ since it is where Γ_0 is defined. We use a SVD to extract the singular values of M:

$$M = V \Sigma^{\dagger} U^T \tag{6}$$

Where V and U are the orthonormal matrices obtained from SVD and:

$$\Sigma^{\dagger} = \text{diag}\left(\frac{\sigma_1}{\sigma_1^2 + \epsilon_d^2}, ..., \frac{\sigma_{m_l}}{\sigma_{m_l}^2 + \epsilon_d^2}\right) \tag{7}$$

$\sigma_1, ..., \sigma_{m_l}$ are the singular values of $J_d(:, 6:12)$. They must remain positive during the manipulation to avoid instability. ϵ_d represents the damping term introduced in the Jacobian to avoid singular values. This implies that the forces and moments applied by the manipulators counteract the elastic restoring forces applied by the object in $s = 0$, thus ensuring the local stability around the equilibrium point where J_d was computed. We therefore define the stable neighborhood $\Omega : \{\Gamma_{0,k+1} \in \mathbb{R}^{12} \mid \min([\sigma_1, ..., \sigma_{m_l}]) > 0\}$ in which each new Γ_0 must be computed.

V. Experimental Validation

A. Experimental Setup

To validate our approach, we used two Franka Emika 7 DOF robots to manipulate various DLOs with different properties:

- Object 1: A squared section DLO made of rubber with homogeneous and isotropic properties and high flexibility.
- Object 2: A braided steel cable with a circular section and very low flexibility. The material is neither homogeneous nor isotropic.
- Object 3: A braided steel sheathed cable with a circular section and a smaller radius giving it higher flexibility. The cable is made of composite materials, not homogeneous nor isotropic either.

Fig. 3: Different objects used through the experiments

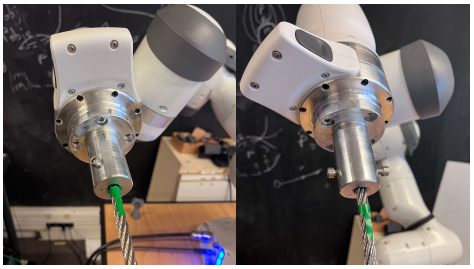


Fig. 4: Fixation of the cable before (left) and after the manipulation (right)

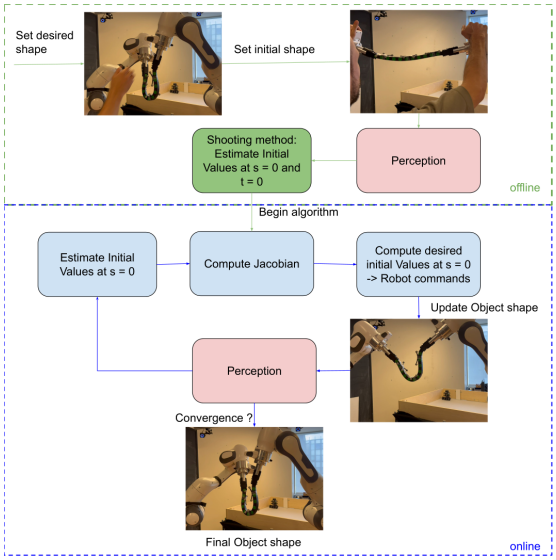


Fig. 5: Experimental protocol

The three manipulated objects can be seen in figure 3. For the vision feedback, we use a set of calibrated Optitrack cameras to track markers placed on the object's control points. The approach is modulable, any vision approach can be used. The control points must effectively capture the primary deformation modes of the object, meaning their number depends on the object's rigidity and length. For the elastic band, we use three markers, while for the longer cables, we use four. Their exact positions are not critical and vary during our tests, provided they are distributed across the entire object. We decided to use the Optitrack system both for its accuracy and robustness to occlusions.

We use a computer with an Intel Xeon Silver 4214R CPU and the standard chipset GPU. We use ROS to communicate with the robot and the algorithm seamlessly operates at a frequency of 10 Hz, where 90 percent of the computational workload is due to the vision component. We also designed two grippers to comply with the clamped assumptions at the boundaries, i.e. no relative displacement at the contact points. While the conditions are respected for objects 1 and 3, there is slippage with object 2. Its effects are however compensated by the controller as the relative positions of the control points remains consistent. The slippage during the experiments ranged between 1 to 2 centimeters along the object's axis, and up to $\pi/2$ around the axis (see figure 4). If the slippage were more significant, we could track the end-effector's position during manipulation and estimate the slippage amplitude by comparing the measured relative positions of the control points and the end-effector with their predicted positions. Once estimated, the slippage amplitude can be used to adjust both the boundary values of the problem and the object's length in real time.

B. Protocol and initial object configuration

To evaluate the efficacy of our controller, we conducted three experiments, all involving three-dimensional shape control tasks. The first experiment focuses on bi-arm three-dimensional shape control of a DLO. We established an objective shape for the object and tasked the controller with achieving it from the initial configuration of the object. The second experiment assessed the repeatability of the approach by deforming the object towards a single predefined shape across five consecutive trials, each with a subtly varied initial object configuration. The final experiment resembles the first one, but with altered boundary conditions. In this case, the manipulated DLO is clamped at one end and held by the robot at the other end.

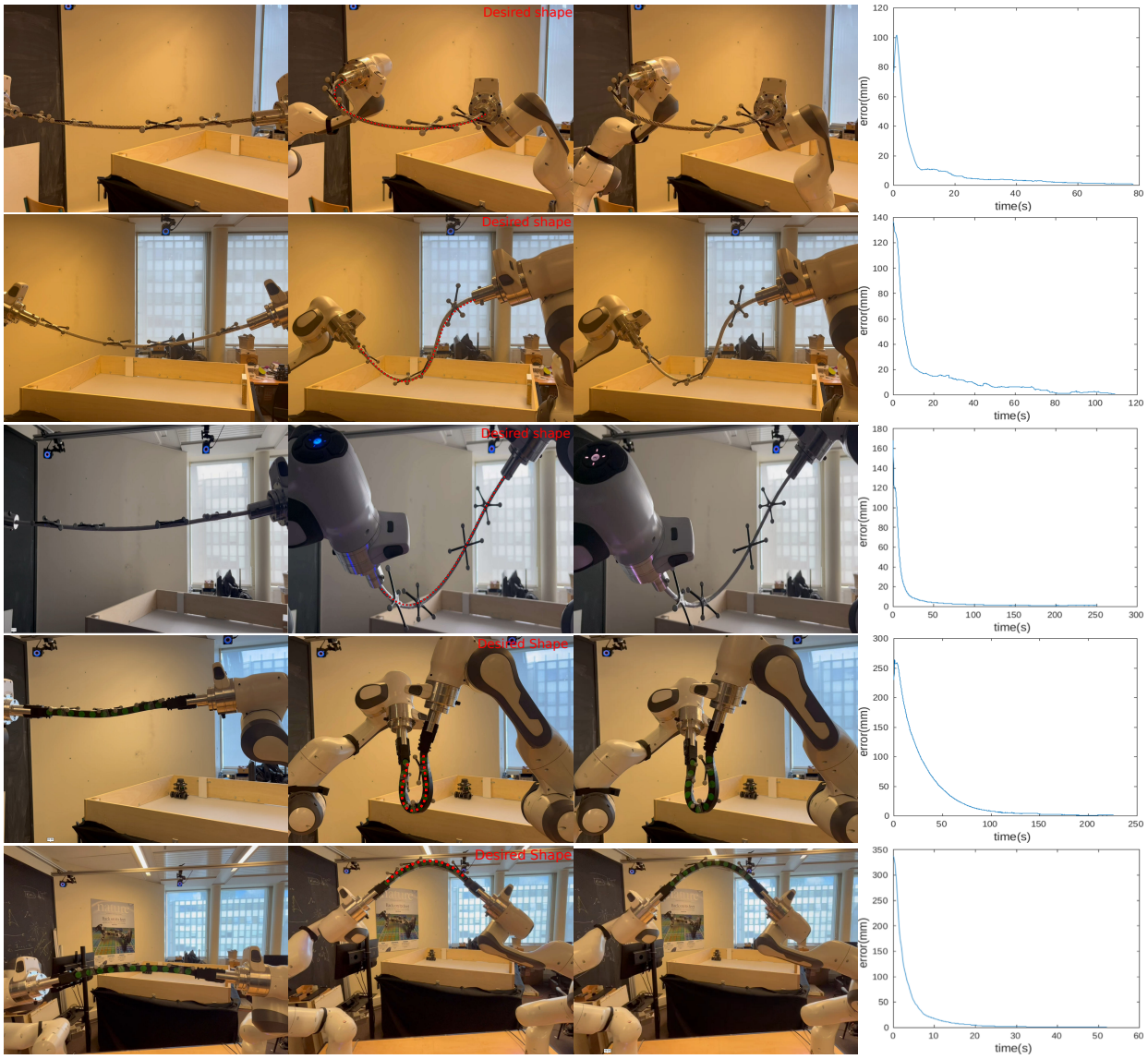
We propose the following protocol, represented in figure 5. First, we define the set of objective points p^i_{obj}, which represent the desired shape of the object. Next, we set the robot's end-effectors at a random initial location, activate the controller, and record the position error ϵ_p over time. The initial position of the end-effectors determines the starting configuration of the manipulated object. The algorithm can begin from any initial object configuration, utilizing any method that estimates or measures the initial values Γ_0 at $t = 0$ corresponding to the object's initial shape. For each test, we manually set the initial position of the end-effectors using the guiding mode of the Franka Emikas. Subsequently, we use an algorithm based on the shooting method to estimate the object's configuration. Typically, the estimation process using the shooting method takes between one and two seconds. As a result, each initial configuration varies slightly from test to test. Almost all cases of failure come from a poor estimation of Γ_0 at $t = 0$, since the shooting method provides accurate estimations for the position and orientation, but is way less accurate when it comes to internal forces and moments.

C. First experiment: bi-arm shape control

We tested our protocol on the three objects (see figure 3) with different objective shapes. A comparison between the initial shape, final shape, and desired shape as well as the error ϵ_p over time is displayed in figure 6 for a sample of all the tests. The error over time curves all have an exponential profile where most of the deformation happens in the first seconds.

The elastic parameters have been estimated for object 1 (Young modulus $E = 3.2MPa$ and Poisson ratio $\nu = 0.5$). The steel modulus for the second and third objects are $E = 180GPa$ and Poisson ratio $\nu = 0.303$. However, deriving the stiffness matrices analytically for the cables proves challenging due to their mechanical characteristics. Therefore, we pre-estimate these matrices by comparing the actual deformation with predictions from the model. As a result, the obtained matrices are somewhat approximate and depend on the object's configuration.

The requirement for precise estimation of stiffness parameters is a primary limitation in physics-based models. Our approach effectively addresses this issue by maintaining consistent precision, regardless of the exactness of these parameters. The majority of errors stemming from inaccuracies in stiffness parameters nullify each other during Jacobian calculation using finite differences. The remaining residual error solely reflects the modeling inaccuracies introduced by



Initial Shape Final Shape Desired shape Error over time

Fig. 6: Comparison between the initial, final and desired shapes. All desired shapes are set in three dimensions. The error over time is displayed in the left column and converges under one mm.
Video link: https://www.youtube.com/watch?v=UtURz1ssXO0&

the disturbance Δ_d^j which is negligible. We conducted a total of 35 tests with different desired and initial shapes. The final errors ϵ_p, obtained after convergence when the algorithm is stopped, are displayed in figure 7. Out of our 35 tests, only 6 converge with a final error ϵ_p larger than 1 mm. We also had 4 cases of failure. We considered a test failed when the final error was larger than 3 mm. The causes for these cases of failure were either a poor estimation of Γ_0 at $t = 0$ or an unattainable objective shape.

D. Second experiment: repeatability

To test the repeatability of our approach, we defined a target shape and attempted to reach it in five consecutive tests. Since the initial shape of the object is set by hand, it is slightly different every time and the error ϵ_p at $t = 0$ varies between 90 and 160mm among the successive tests. The final shape, as well as the error ϵ_p for these five repeated tests are represented in figure 8. The profile of the error curves and the final shapes are very similar for each test. Except for the 4th test, the object converges toward the desired shape with a sub-millimeter error at every test, regardless of the initial shape of the object.

E. Third experiment: different boundary conditions

We followed the same protocol as proposed in experiment 1, however in this case we only had one arm manipulating the object, which is clamped at the other limit. This changes the boundary conditions of the problem, thus adding additional constraints on the position and orientation of the object at

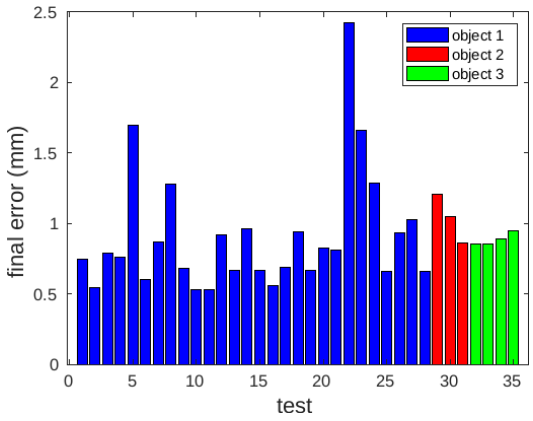


Fig. 7: Final error after convergence for every test

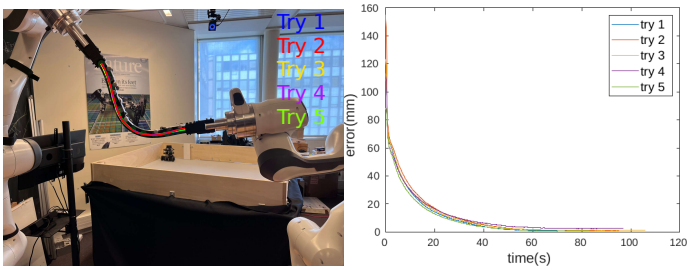


Fig. 8: Repeatability: Final shapes and error over time of five successive tests

$s = 0$. The disturbance Δ_d^j introduced along these dimensions in Γ_0 will therefore be equal to zero when computing the Jacobian. To avoid computing the pseudo-inverse of sparse matrices, we reduce the dimension of the problem to the sole variation of n_0 and m_0 in Γ_0. This effectively reduces the number of variables by half. The comparison between the initial, final, and desired shapes is represented in figure 9.

F. Comparison with the state of the art and Discussion

A comparison of the most promising approaches including open-loop Cosserat for robotic shape control of linear deformable objects is displayed on table I. We evaluate several factors: from the physical realism of the obtained shape (which ensures feasibility) to the shape complexity corresponding to the highest order of the Legendre polynomial that approximates the proposed shapes. The only area where our algorithm falls behind is the loop frequency which can be easily sped up by changing the optitrack system with another vision-based approach (for reference the loop frequency without the vision is 100 Hz). On the other side, our approach is the only one that consistently reaches sub-millimeter precision for such a range of complex desired shapes, even when manipulating objects with complex compositions and unknown properties.

The closest method in term of performance is the ARAP-SS algorithm proposed in [14]. To further enhance the comparison, we introduce the same normalized metric based on cosine similarity between the measured positions of the control points and their predicted positions. This metric represents the alignment between the predicted and measured vectors. The nearer the value is to 1, the more accurately the deformation Jacobian reflects the behavior of the manipulated object. The mean between the cosine similarities of each tests as well as the inter-test standard deviation is exposed on table II.
Our approach shows higher mean cosine similarity for all objects than the ARAP-SS method. The standard deviation can directly be tied to the robustness of the approach as it shows a regularity of the metric across different tests. The difference in mean and standard deviation between the first object and the others reflects the imprecision on the stiffness parameters. Nevertheless, we can conclude that the Jacobian accurately represents the behavior of the manipulated object in all scenarios. The Jacobian primarily captures the direction of deformations, which necessitates precise modeling. On the other side, the stiffness parameters mainly influence the amplitude of said deformations, which can be readily adjusted through closed-loop control. This explains the robustness of the approach to stiffness estimation.

However, the method has limitations. Firstly, although it is general amond DLOs, the Cosserat rod model used in this article is specifically designed for one-dimensional objects and is therefore not suitable for modeling 2D or 3D shapes. Secondly, the method does not accommodate unforeseen contacts that have not been predefined in the model. Managing unexpected contacts remains an open challenge in the field of deformable object manipulation.

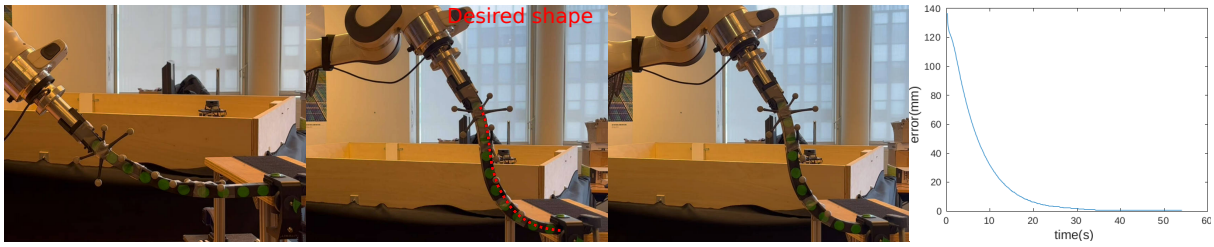
VI. Conclusion

We have presented in this article our method to control the shape of DLOs in real-time. By defining our global BVP and iteratively solving an IVP, we included the Cosserat rod model in our closed-loop controller. Thus enhancing the precision compared to regular open-loop physics-based approaches while also demonstrating robustness to stiffness estimation. We then conducted three different experiments, showcasing the precision and repeatability of the approach for various desired shapes. The approach consistently achieved sub-millimeter precision during shape control tests on various objects with anisotropic and heterogeneous properties. An interesting perspective would be to replace the perception module with more advanced methods to detect and estimate contacts accurately. The contact forces could then be included in the model to allow the approach to handle unexpected contacts. Finally, the approach can be generalized to other applications such as deformable linear robots control. The problem's formulation varies depending on the type of actuation used. However, it is generally simpler because the position and orientation are fixed at the robot's base. Consequently, the main challenge is accurately defining the relationship between the variation of internal forces and moments at $s = 0$ and the actuation.

References

[1] 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.

[2] 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.



Initial Shape Final Shape Desired shape Error over time

Fig. 9: Experiment with different boundary conditions: The object is grasped at one limit and clamped at the other

Method	Physically realistic	Precision	Convergence	Loop frequency	Shape complexity	Material	Object type
O-L Cosserat [23]	Yes	5 mm	Linear	Open-loop	3	Isotropic+homogeneous	Linear
ARAP-SS [14]	No	5 mm	Linear	20 Hz	3	Isotropic+homogeneous	Linear+Planar
DLO NN [26]	No	10 mm	Linear	1500 Hz	3	Isotropic+homogeneous	Linear
FEM [4]	Yes	20 mm	Exponential	30 Hz	2	Isotropic+homogeneous	Linear
Our approach	Yes	< 1 mm	Exponential	> 10 Hz	3	Any	Linear

TABLE I: Comparison of the shape servoing of deformable linear objects (DLO) methods

Method	Our (object 1)	Our (object 2)	Our (object 3)	ARAP
mean	0.9884	0.9792	0.9771	0.8483
std	0.0082	0.0349	0.0154	0.0379

TABLE II: Average cosine similarity and standard deviation over all tests for each object and method

[3] 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.

[4] A. Koessler, N. R. Filella, B.-C. Bouzgarrou, L. Lequièvre, and J.-A. C. 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)*. IEEE, 2021, pp. 1637–1643.

[5] 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.

[6] 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.

[7] 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.

[8] 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.

[9] 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.

[10] 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.

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

[12] F. Alambeigi, Z. Wang, R. Hegeman, Y.-H. Liu, and M. Armand, "Autonomous data-driven manipulation of unknown anisotropic deformable tissues using unmodelled continuum manipulators," *IEEE Robotics and Automation Letters*, vol. 4, no. 2, pp. 254–261, 2018.

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

[14] 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.

[15] O. Aghajanzadeh, G. Picard, J. A. C. Ramon, C. Cariou, R. Lenain, and Y. Mezouar, "Optimal deformation control framework for elastic linear objects," in *2022 IEEE 18th International Conference on Automation Science and Engineering (CASE)*. IEEE, 2022, pp. 722–728.

[16] F. Boyer, V. Lebastard, F. Candelier, and F. Renda, "Dynamics of continuum and soft robots: A strain parameterization based approach," *IEEE Transactions on Robotics*, vol. 37, no. 3, pp. 847–863, 2020.

[17] F. Boyer, V. Lebastard, F. Candelier, F. Renda, and M. Alamir, "Statics and dynamics of continuum robots based on cosserat rods and optimal control theories," *IEEE Transactions on Robotics*, vol. 39, no. 2, pp. 1544–1562, 2022.

[18] 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.

[19] 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.

[20] 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.

[21] S. S. Antman, "Problems in nonlinear elasticity," *Nonlinear Problems of Elasticity*, pp. 513–584, 2005.

[22] 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.

[23] A. Azad, H. Quentin, B. A. Faiz, and P. Véronique, "Optimal cosserat-based deformation control for robotic manipulation of linear objects," in *2023 IEEE/ASME International Conference on Advanced Intelligent Mechatronics (AIM)*. IEEE, 2023, pp. 381–388.

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

[25] H. B. Gilbert, D. C. Rucker, and R. J. Webster III, "Concentric tube robots: The state of the art and future directions," in *Robotics Research: The 16th International Symposium ISRR*. Springer, 2016, pp. 253–269.

[26] A. Caporali, P. Kicki, K. Galassi, R. Zanella, K. Walas, and G. Palli, "Deformable linear objects manipulation with online model parameters estimation," *IEEE Robotics and Automation Letters*, 2024.