Robotic co-manipulation of deformable linear objects for large deformation tasks

Karam Almaghout, Andrea Cherubini, Alexandr Klimchik

Highlights

Robotic Co-manipulation of Deformable Linear Objects for Large Deformation Tasks

Karam Almaghout,Andrea Cherubini,Alexandr Klimchik

- We address the problem of complex deformation of DLO and propose a new approach to handle it
- We evaluate the proposed approach in both simulation and real-life experiments.
- We conduct a comprehensive review, along with a comparative study on the most recent works in the literature in the DLO shape control and the techniques for modeling DLOs.

Robotic Co-manipulation of Deformable Linear Objects for Large Deformation Tasks

Karam Almaghout[a,*], Andrea Cherubini[b] and Alexandr Klimchik[c]

[a]Intelligent Robotics Systems Laboratory, Institute of Robotics and Computer Vision, Innopolis University, Universitetskaya St, 1, Innopolis, 420500, Russia

[b]LIRMM, Universite de Montpellier, CNRS Montpellier, 34090, France

[c]Lincoln Centre for Autonomous Systems (L-CAS), School of Computer Science, University of Lincoln, Lincoln, 695014, United Kingdom

ARTICLE INFO

Keywords:
Robotic co-manipulation
Deformable objects manipulation
Optimization control
Deformable objects modeling

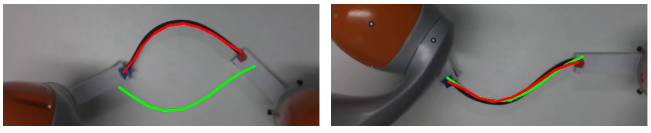
ABSTRACT

This research addresses the challenge of large/complex deformation in the shape control tasks of Deformable Linear Objects (DLO). We propose a collaborative approach using two manipulators to achieve shape control of a DLO in 2D workspace. The proposed methodology introduces an innovative Intermediary Shapes Generation (ISG) algorithm which outputs a series of intermediary shapes to guide the DLO towards the desired shape. The robot controller is formulated as an optimization problem, where the main objective is to minimize the error between the current shape and the desired shape, while ensuring the diminishing rigidity property of the DLO as a constraint. We conduct extensive simulations and real-life experiments to evaluate the effectiveness of our approach. We consider various scenarios of basic shapes, as well as complex deformations with opposite concavities between initial and final shapes. The outcomes demonstrate the robustness and high accuracy of the proposed system, in achieving complex deformations. This capability represents the primary contribution of our research. The optimization-based control framework, coupled with the ISG algorithm, enables effective shape control without the need for extensive modeling nor training, and offers a promising solution for practical applications requiring precise shape control of DLOs. Moreover, we carry out a thorough review and comparative analysis encompassing the latest literature in DLO shape control, and the techniques for DLO modeling.

1. Introduction

Deformable linear objects (DLOs), such as cables, ropes, and sutures, are widely used in industry, medicine, and everyday life. However, the problem of DLOs handling and manipulation is considered as a bottleneck of automation and robotics (Trommnau, Kühnle, Siegert, Inderka and Bauernhansl (2019); Heisler, Steinmetz, Yoo and Franke (2017)). This has led to a growing interest in DLO robot manipulation, e.g., for cable routing, shape control of a rope, and surgical suturing. Although great progress has been achieved in recent years, shape control remains an open problem in robotics (Zhu, Cherubini, Dune, Navarro-Alarcon, Alambeigi, Berenson, Ficuciello, Harada, Kober, Li et al. (2022)).

The shape control tasks of DLOs can be divided into two types. The first type concerns manipulating soft DLOs placed on tables: the robot can move the DLO to the desired shape by executing a series of grasp-and-release actions at different points of the DLO (Yan, Vangipuram, Abbeel and Pinto (2021); Zhang, Schmeckpeper, Chaudhari and Daniilidis (2021)). The second type is the manipulation of stiffer DLOs such as cables, for which deformation under forces is mainly elastic (Laezza and Karayiannidis (2021); Jin, Wang and Tomizuka (2019)). The robot grasps only the DLO ends to control its shape, either in 2D or 3D workspace. In this paper, we focus on the second type of shape control tasks: manipulation of elastic DLO grasped at its two ends by two manipulators in 2D workspace. The manipulators guide the DLO towards a desired shape based on visual servoing. While most existing works consider local and small deformation, this work addresses the challenge of large deformation: the initial and desired shapes are quite different from each other and may even feature opposite concavities. Figure 1 shows an example of the considered cases. To the best of the authors' knowledge, this is the first time this type of deformation is explicitly addressed in the robotics literature.



(a) Initial shape (b) Final Shape

Figure 1: An example of a large deformation task. The DLO (red) is concave upwards while the desired shape (green) is concave downwards

To deform it accurately, the DLO shape must be tracked throughout the task, by a proper sensory system, including vision, force, and/or tactile sensors (Delgado, Corrales, Mezouar, Lequievre, Jara and Torres (2017); Sanchez, Mohy El Dine, Corrales, Bouzgarrou and Mezouar (2020)). Among these, vision sensors, i.e., cameras, are widely used since they are affordable and appropriate for tasks

*Corresponding author
k.almaghout@innopolis.university (K. Almaghout); andrea.cherubini@lirmm.fr (A. Cherubini); AKlimchik@lincoln.ac.uk (A. Klimchik)
ORCID(s):

which include detection and localization of both rigid and deformable objects (Sanchez, Corrales, Bouzgarrou and Mezouar (2018); Almaghout, Boby, Othman, Shaarawy and Klimchik (2021)). In the vision-based robotic manipulation of DLOs, some works use marked DLOs. These marks are considered as feature points, and the robot manipulates these points towards desired positions (Tang, Wang and Tomizuka (2018); Li, Wang and Liu (2019); Yan et al. (2021)). A significant limitation of this approach is that it is not realistically possible to add these markers on every DLO to be manipulated. Others consider the DLO contour as the DLO feature (Zhu, Navarro, Fraisse, Crosnier and Cherubini (2018); Zhu, Navarro-Alarcon, Passama and Cherubini (2021)). This approach is more feasible than the previous one, but it requires more calculations, since it handles a greater number of points (the contour points). In our previous work (Almaghout and Klimchik (2022b)), we developed a virtual feature points (VFPs) algorithm, which tracks the DLO by representing it as a set of virtual points. This algorithm eliminates the need for physical markers and reduces the computational cost needed to process the contour. Once we obtain the DLO feature points, the desired shape is generated to have the same number of points.

In this work, we propose a new algorithm for intermediary shapes generation (ISG). This ISG algorithm takes the initial and desired shapes (each represented as a set of points), and generates intermediary shapes which can be considered as local desired shapes. Once these intermediary shapes are generated, the robots guide the DLO from the current shape to the desired shape through these intermediary ones. The robots motion is planned as an optimization control problem (OCP), where the desired displacement of the robot end-effectors are computed to minimize the error between the current DLO shape and the next intermediary shape, in an iterative manner towards the desired shape. Figure 2 shows an overview of the proposed scheme. The proposed approach and algorithms are evaluated in simulation and real experiments for different shapes. The main contributions of this paper are summarized hereby.

1. We tackle the problem of complex deformation of DLO, including opposite-concavity deformation, i.e., the desired shape has a concavity that is the reverse of the initial one.
2. We conduct a review and comparative analysis of the latest works in the field of DLO manipulation, including various techniques for modeling DLOs.
3. We propose a novel algorithm (ISG) for generating intermediary shapes, to handle large and complex deformation scenarios.
4. We formulate the shape control task as an optimization control problem, considering the diminishing rigidity property of DLO.
5. We conduct simulations and real experiments to validate our approach in various cases.

To address these problems, the rest of the paper is organized as follows. In the next section, we review the most recent works in the robotics literature. Section 3 states the problem of interest and defines some notations. The ISG algorithm is introduced in section 4. Section 5 discusses the robots motion planning problem. In section 6, the DLO dynamic modeling techniques are discussed and the mass-spring model is exerted for simulation study, which is developed in section 7. In section 8, real experiments are presented and discussed. Finally, we conclude and discuss the future works in Section 9.

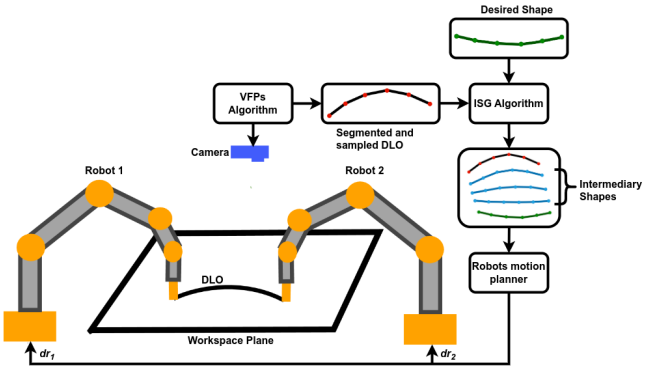


Figure 2: The proposed framework. The virtual feature points (VFPs) algorithm takes the captured frame, segments the DLO, and samples it into a set of feature points. The intermediary shapes generation (ISG) algorithm takes the sampled and desired shapes and generates intermediary shapes. The DLO moves towards the desired shape through the intermediary shapes by the guidance of the robots, which receive the desired displacements as the output of the motion planner.

2. Related Works

In this section, we begin with a thorough review and comparison of the techniques employed for modeling and simulating DLOs. Subsequently, we categorize and discuss the various approaches used for DLO manipulation, while also providing a comparison of the latest works in this area.

2.1. DLO Modeling

The problem of simulation and modeling of DLOs is crucial for many applications: in virtual reality, for training and assessing suturing skills Xu and Liu (2018), in reinforcement learning to train robots handling these objects Laezza and Karayiannidis (2021), and in parameter identifications of real DLOs to improve the manipulation accuracy Liu, Su, Lu, Li and Yip (2022). In contrast to rigid objects, it is challenging to obtain the exact models of DLOs; they can be hardly calculated theoretically, and modeling parameters may vary significantly among DLOs. Several techniques have been developed in the literature to model flexible cables and other DLOs. These include: mass-spring model (Lv, Liu, Ding, Liu, Lin and Ma (2017)), multi-body model (Servin and Lacoursiere (2008)), position-based dynamics model (Xu and Liu (2018)), elastic rod model(Linn and Dreßler (2017)), and dynamic spline model (Valentini and Pennestrì

Table 1
Characteristics of DLO modeling techniques

Model	*Advantages*	*Limitations*
Mass-Spring	➤ Widely used and easy to implement. ➤ Computationally efficient, leading to fast results. ➤ Requires less memory usage.	➤ The accuracy is limited in large deformation.
Multi-body	➤ Intuitive. ➤ Good real-time performance. ➤ Requires less memory.	➤ The accuracy is limited in large deformation.
Position-based Dynamics	➤ Fast and stable performance.	➤ Lacks of physical explanation. ➤ Visual reliability only.
Finite element	➤ High accuracy. ➤ Realistic behavior.	➤ Associated with significant computational resources and cost.
Elastic Rod	➤ Good theoretical basis. ➤ Realistic and accurate behavior.	➤ Associated with significant computational resources and cost. ➤ Slow computation speed.
Dynamic Spline	➤ Good theoretical basis. ➤ Continuous model.	➤ Relatively high computational resources and cost. ➤ Slow computation speed.

(2011)). Different modeling techniques were analyzed by Lv, Liu, Xia, Ma and Yang (2020), and Yin, Varava and Kragic (2021) have conducted in-depth analyses of these modeling techniques. The authors have provided valuable insights into the strengths and limitations of each technique. These findings assist in determining the most suitable modeling methods for DLOs, considering their specific behaviors and characteristics. The overview of different DLO's modeling techniques and their characteristics are given in Table 1. The table sums up key advantages and limitations of each approach. In addition, Table 2 compares the approaches, based on the following indicators:

- *Complexity*: indicates the complexity of implementing and understanding the model.
- *Computational cost*: refers to the computational resources and time required to simulate the model.
- *Resources cost*: reflects the amount of memory and other resources needed to run the model.
- *Accuracy*: represents the model's ability to accurately simulate deformations.
- *Real-time applications*: indicates whether the model can be used in real-time applications, where timely responses are crucial.
- *Stability*: represents the model's ability to handle various input scenarios and maintain stability in complex deformation scenarios.
- *Versatility*: indicates the model's flexibility in simulating different types of objects dynamics.

Based on this overview and comparison, it is shown that the mass-spring model provides a reasonable trade-off between model accuracy, implementation complexity, and computational cost; this motivates us to to use it as a primary modeling approach for simulation evaluation.

2.2. DLO Manipulation

In recent years, there has been significant research on robotic shape control of DLOs. Herein, we review some of the recent works in this field and categorize them into four main groups based on the DLO manipulation control approach:

1. Model-based approach.
2. Jacobian-based approach.
3. Data-driven approach.
4. Hybrid approach.

Let us analyze their particularities and advantages in detail, in the following sections.

Table 2
Comparison of DLO modeling techniques

Model	Complexity	Computational Cost	Resources Cost	Accuracy	Real-time application	Stability	Versatility
Mass-Spring	Low	Low	Low	Medium	Yes	Limited	Medium
Multi-body	Medium	Low	Low	Medium	Yes	Limited	Medium
Position-based Dynamics	Medium	Low	Low	Medium	Yes	Medium	High
Finite element	High	High	High	High	No	High	High
Elastic Rod	High	High	Medium	High	No	High	High
Dynamic Spline	Medium	Medium	Medium	Medium	No	Limited	Medium

2.2.1. Model-based approach

In the model-based approaches, the physical model of the DLO is used to predict the behavior of how it will deform in response to different manipulation strategies, see Figure 3. The model is used to plan the manipulation, in order to deform the DLO into the desired shape. Duenser, Bern, Poranne and Coros (2018) develop a finite element model (FEM) simulation of DLO for open-loop shape control. An approach using reduced FEM to closed-loop shape control of DLOs is proposed by Koessler, Filella, Bouzgarrou, Lequièvre and Ramon (2021). This approach is limited by the accuracy of the model used for it, as well as by the complexity of the DLO's behavior, which makes obtaining the accurate model difficult in practice. Liu et al. (2022) propose a model-based shape control approach. In their approach, they consider position-based dynamics of DLO. They estimate the model parameters by minimizing the shape error between the real DLO and the simulated model. Model-based approach proved their efficiency in applications where the accuracy is critical, such as surgical suturing tasks. However, the lack of precise model parameters knowledge is the main drawback of this approach.

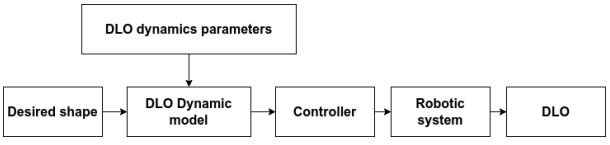
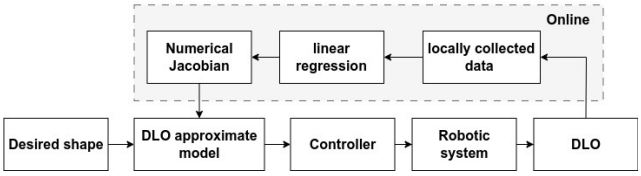


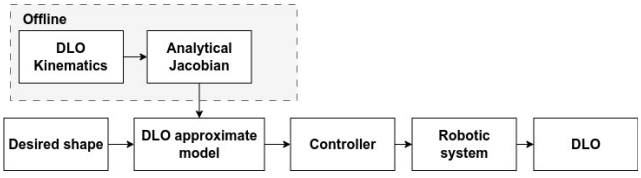
Figure 3: Block diagram of model-based approach for DLO manipulation.

2.2.2. Jacobian-based approaches

These approaches use simplified models of the DLO deformation to compute the required robot control inputs, which can be more computationally efficient than using a full DLO model. Approximation approaches for DLO shape control offer a trade-off between computational complexity and shaping accuracy. Assuming that a small change of the DLO is linearly related to a small displacement of the robot, many works developed a Jacobian-like approximate model, to map the DLO shape to the robot end-effectors motion. This Jacobian is computed numerically during manipulation, as shown in Figure 4a. For instance, Zhu et al. (2018), Jin et al. (2019), and Lagneau, Krupa and Marchal (2020) introduced methods based on online estimation of the local deformation model of DLOs. Another direction consists in formulating the Jacobian analytically, considering DLO properties as well as the manipulation conditions, Figure 4b. In particular, in our previous works, we introduced an approximate model based on the diminishing property of the DLO (Almaghout and Klimchik (2022a,b)). Ruan, McConachie and Berenson (2018) developed a new geometric model, based on the directional rigidity and constraints. Although these approaches are simple to implement, they are not suitable for highly deformable DLOs, and fail in large deformation tasks.



(a) Numerical Jacobian



(b) Analytical Jacobian

Figure 4: Block diagram of Jacobian-based approach for DLO manipulation.

2.2.3. Data-driven approaches

In data-driven approaches, the robot learns to control the shape of the object either through offline training, demonstration, or trial-and-error, see Figure 5. Nair, Chen, Agrawal, Isola, Abbeel, Malik and Levine (2017), Tang et al. (2018), and Yan, Zhu, Jin and Bohg (2020) present learning-based approaches where the robot observes how a human deforms the DLO, and then imitates the observed behaviors.

Reinforcement learning (RL) is another direction where the robot learns a policy, by the rewards collected during the interaction. RL policies are learned for shape control of elastoplastic DLOs in (Laezza, Gieselmann, Pokorny and Karayiannidis (2021)) and (Zakaria, Aranda, Lequièvre, Lengagne, Ramón and Mezouar (2022)). Zakaria et al. (2022) develop a new framework for shape control of DLOs, using Deep RL. Huang, Xia, Wang and Liang (2023) propose a new framework for DLO shape control. They suggest a new approach to control the shape of DLO using external contacts, to supplement the control system and compensate for its underactuation. The suggested framework includes a learning module named Learning Graph Dynamics with External Contact (LG-DEC), for predicting future outcomes and an action generation module for manipulating the DLO in an iterative manner. The LG-DEC module is trained by data collected in a simulation environment. The action generation module outputs optimal actions through random sampling, forward prediction, and SGD optimization. These approaches can be particularly effective for complex DLOs, that are difficult to model accurately, but require a large amount of training data and can be computationally expensive.

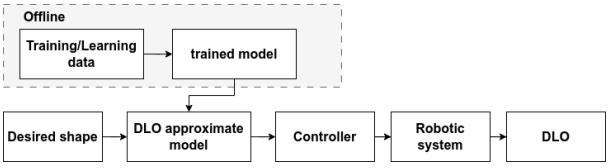


Figure 5: Block diagram of data-driven approach for DLO manipulation.

2.2.4. Hybrid approaches

Some recent studies have proposed hybrid approaches, which combine a model-based or Jacobian-based approach augmented with data-driven approach, see Figure 6. For example, Wang, Zhang, Zhang, Wu, Zhu, Jin, Tang and Tomizuka (2022) developed a hybrid offline-online method to learn the dynamics of DLOs. A graph neural network (GNN) is utilized to learn the deformation dynamics from the simulation data. Then, a linear residual model is learned in real-time to bridge the sim-to-real gap. The learned model is then used as the dynamics constraint of a trust region-based model predictive controller, to calculate the robot desired motion. Yu, Zhong and Li (2022) proposed a new scheme for DLOs shape control, where the initial model is approximated as a local linear model by a neural network trained offline. The approximated model is used in an adaptive controller to achieve the shape control task. The neural network is further updated online to compensate for any errors in the offline model caused by insufficient training or by changes of DLO properties. Although these approaches offer more accurate performance for complex shapes, they suffer from the training expense, and require a sufficient amount of data earlier.

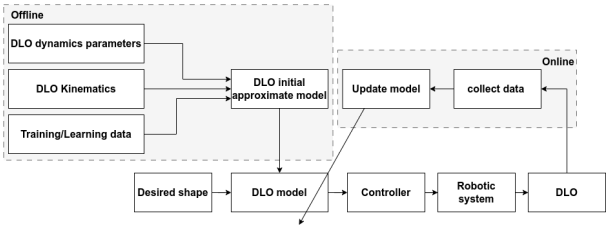


Figure 6: Block diagram of hybrid approach for DLO manipulation.

2.3. Summary

Overall, each of the modeling and manipulation approaches mentioned above has its advantages and limitations, imposing trade-offs for specific applications, depending on the desired properties and accuracy. Table 3 provides a concise summary of the general advantages, limitations, and applications of the approaches we have reviewed. In addition, Table 4 offers an overview of the most recent works related to these approaches with some details on the model particularities and use cases.

In this work, we address the large/complex deformation challenge. The large deformation is solved mainly by a novel intermediary shapes generation (ISG) algorithm that allows the system to guide the DLO to the desired shape. The control is defined as an optimization problem. A Jacobian-based approximate model based on the diminishing rigidity property of the DLO is defined as a constraint. The loss function describes the deflection between the DLO current shape and desired one. No prior knowledge of the DLO dynamics and characteristics is required. Thus, this system will have the Jacobian-based advantages and overcome its limitations to solve the large deformation problem. Further, we use the DLO mass-spring model (for its simple implementation), in simulations, for evaluation purposes.

3. Problem Formulation

Let us consider two robot arms rigidly grasping a DLO at its two ends. The robots cooperatively manipulate the DLO on a 2D plane to move it from an initial to a desired shape. The DLO initial and desired shapes are represented as a set of N points uniformly distributed along the DLO length. The distance between each pair of adjacent points is constant l_s. Figure 7 shows a DLO grasped by two end-effectors along with the desired shape.

Let $\boldsymbol{c} = [\boldsymbol{c}_1,\ \boldsymbol{c}_2,\ \cdots,\ \boldsymbol{c}_N]^T$ and $\boldsymbol{s} = [\boldsymbol{s}_1,\ \boldsymbol{s}_2,\ \cdots,\ \boldsymbol{s}_N]^T \in \mathbb{R}^{2N}$ be the vector of DLO point coordinates and desired shape point coordinates, respectively; where $c_n = [c_{nx},\ c_{ny}]^T$ and $s_n = [s_{nx},\ s_{ny}]^T$, for $n = 1, 2, ..., N$. The robots' end-effectors configurations are defined by vector $\boldsymbol{r} = [\boldsymbol{r}_1,\ \boldsymbol{r}_2]^T \in \mathbb{R}^6$; where $r_m = [r_{mx},\ r_{my},\ \varphi_m]^T$, for $m = 1$, or 2.

The problem is to design the robots controller to guide the DLO towards its desired shape. In the literature, researchers have defined Jacobian-like models to map the motion of the robots to that of the DLO. However, these

Table 3
Summary DLO manipulation control approaches

Approach	*Advantages*	*Limitations*	*Applications*
Model-based	➢ Accurate predictions. ➢ Precise manipulation. ➢ Can handle nonlinear behavior locally.	➢ Difficult to obtain an accurate model. ➢ Sensitive to parameter uncertainties. ➢ High computational cost.	Suitable for applications where accurate modeling is critical and deformations are not highly complex.
Jacobian-based	➢ Computationally efficient. ➢ Suitable for real-time control.	➢ Limited accuracy for large deformations or nonlinear behavior.	Suitable for applications where real-time control is critical and deformations are not highly complex.
Data-driven	➢ Can adapt to changes in behavior during large and complex deformation. ➢ Robust to modeling errors and parameters uncertainties.	➢ Requires a large amount of training data. ➢ Accuracy depends on quality and quantity of data. ➢ Requires more computational resources. ➢ Generalization to different untrained DLOs cannot be guaranteed.	Suitable for applications with complex and large deformation.
Hybrid	➢ Combines strengths of multiple approaches. ➢ Can handle modeling errors and uncertainties. ➢ Can adapt to changes in behavior.	➢ High computational resources and cost. ➢ Complex to implement. ➢ Requires acquiring data (offline and online), and the performance depends on the quality of the acquired data.	Suitable for applications with complex and large deformations, where accuracy is critical.

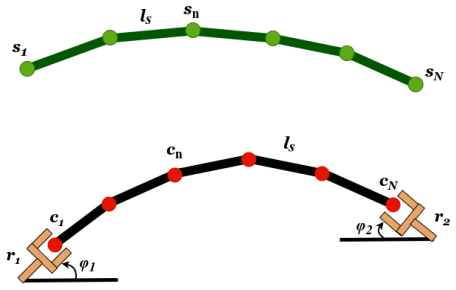


Figure 7: A DLO (black) grasped by two end-effectors (orange) with the desired DLO shape (green).

researches considered relatively straightforward scenarios, where the desired shape $\boldsymbol{s}$ and the initial shape of the DLO $\boldsymbol{c}^0$ are relatively simple and close to each other. These scenarios known as local deformation and do not require large or complex deformation (Qi, Ma, Zhou, Zhang, Lyu and Navarro-Alarcon (2022)) and (Zhu et al. (2018)). Our work addresses different shape control scenarios, where large or complex deformations are required. For instance, the scenarios where the concavity of the initial and desired shapes are opposite.

The robots' motion is planned in two distinct phases. In the first phase, a series of intermediary shapes is created, connecting the initial shape to the desired one. These shapes act as waypoints for each DLO point, guiding each one from its initial position towards the desired one. The primary purpose of these shapes is to simplify the challenges posed by large and complex deformations. The second phase involves manipulating the DLO from its initial shape towards the desired one, following the trajectory defined by the generated intermediary shapes.

4. Intermediary Shapes Generation (ISG)

The first phase of the approach is to generate a set of intermediary shapes representing how the DLO will deform to reach the desired shape. The aim of these shapes is to simplify the large/complex deformation, known as global deformation, into a series of local deformation problems, handled sequentially.

Let $\boldsymbol{c}^0$ and $\boldsymbol{s}$ be the DLO initial and desired shapes, respectively. First, we compute the difference between each DLO point $\boldsymbol{c}^0$ and its corresponding desired point $\boldsymbol{s}$ as follows:

$$\boldsymbol{\delta} = \boldsymbol{s} - \boldsymbol{c}^0 \tag{1}$$

Table 4
Comparison analysis of different work for DLO manipulation

	Manipulators		*Workspace*		*Approach*				*Type of Study*	
	Single	Multiple	2D	3D	Model-based	Jacobian-based	data-driven	Hybrid	Sim.	Real
Nair et al. (2017)	✓		✓				✓			✓
Xu and Liu (2018)	✓			✓	✓				✓	
Tang et al. (2018)		✓	✓				✓			✓
Duenser et al. (2018)		✓		✓	✓				✓	✓
Ruan et al. (2018)		✓		✓		✓			✓	✓
Zhu et al. (2018)		✓	✓			✓				✓
Jin et al. (2019)		✓	✓			✓				✓
Lagneau et al. (2020)		✓		✓		✓				✓
Yan et al. (2020)	✓		✓				✓			✓
Koessler et al. (2021)	✓		✓		✓					✓
Wang et al. (2022)		✓	✓					✓	✓	✓
Almaghout and Klimchik (2022a)		✓	✓			✓			✓	
Almaghout and Klimchik (2022b)		✓	✓			✓				✓
Yu et al. (2022)		✓		✓				✓	✓	✓
Liu et al. (2023)	✓			✓	✓					✓
Huang et al. (2023)		✓	✓				✓		✓	✓

Let us define the DLO point with the maximum distance as $\boldsymbol{c}_\sigma$, with

$$\sigma = \arg\max(\boldsymbol{\delta})\ ;\ \sigma \in \{1, ..., N\}. \tag{2}$$

We also define K, the number of shapes (including the intermediary shapes and the final desired one):

$$K = \lfloor \frac{\boldsymbol{\delta}_\sigma}{\lambda} \rfloor + 1, \tag{3}$$

where λ is a user-defined step of the DLO point $\boldsymbol{c}_\sigma$.

Next, we compute the slope (angle) between each two adjacent points in $\boldsymbol{c}^0$ and $\boldsymbol{s}$, starting from $\boldsymbol{c}_\sigma$ towards both ends, as shown in Fig. 8:

- for each point from $\boldsymbol{c}_\sigma$ and $\boldsymbol{s}_\sigma$ towards $\boldsymbol{c}_N$ and $\boldsymbol{s}_N$, respectively:

$$\theta c_n = \text{atan2}\left((c_{ny} - c_{(n-1)y}),\ (c_{nx} - c_{(n-1)x})\right), \tag{4}$$

$$\theta s_n = \text{atan2}\left((s_{ny} - s_{(n-1)y}),\ (s_{nx} - s_{(n-1)x})\right); \tag{5}$$

- for each point from $\boldsymbol{c}_\sigma$ and $\boldsymbol{s}_\sigma$ towards $\boldsymbol{c}_1$ and $\boldsymbol{s}_1$, respectively:

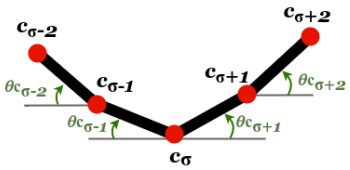


Figure 8: Demonstration of the angles between each two adjacent DLO points in 2D plane.

$$\theta c_n = \text{atan2}\left((c_{ny} - c_{(n+1)y}),\ (c_{nx} - c_{(n+1)x})\right), \tag{6}$$

$$\theta s_n = \text{atan2}\left((s_{ny} - s_{(n+1)y}),\ (s_{nx} - s_{(n+1)x})\right). \tag{7}$$

Then, each angular displacement is given by:

$$\Theta_n = \theta s_n - \theta c_n. \tag{8}$$

We get the angular displacement step size as

$$\vartheta_n^a = \frac{\Theta_n}{K - 1} \tag{9}$$

where ϑ_n^a are the linear and angular step size of the DLO point $\boldsymbol{c}_n$.

Finally, we generate the $K-1$ waypoints for all DLO points starting from $\boldsymbol{c}_\sigma$ in the following order:

$$\boldsymbol{c}_\sigma^\kappa = \boldsymbol{c}_\sigma^{\kappa-1} + \lambda \tag{10}$$

- for DLO points of indices $> \boldsymbol{c}_\sigma$:

$$\boldsymbol{c}_n^\kappa = \boldsymbol{c}_{n-1}^\kappa + \begin{bmatrix} l_s \cos(\theta c_{n-1} + \vartheta_n^a) \\ l_s \sin(\theta c_{n-1} + \vartheta_n^a) \end{bmatrix} \tag{11}$$

- for DLO points of indices $< \boldsymbol{c}_\sigma$:

$$\boldsymbol{c}_n^\kappa = \boldsymbol{c}_{n+1}^\kappa - \begin{bmatrix} l_s \cos(\theta c_{n+1} + \vartheta_n^a) \\ l_s \sin(\theta c_{n+1} + \vartheta_n^a) \end{bmatrix} \tag{12}$$

where the superscript $\kappa = 1, \cdots, K-1$. These waypoints represent intermediary shapes from the initial towards the final desired shape of the DLO $\boldsymbol{s}$, where $\boldsymbol{c}^K = \boldsymbol{s}$. The initial, desired, and intermediary shapes of the DLO are shown in Fig. 9. Algorithm 1 shows a pseudo code of the ISG algorithm.

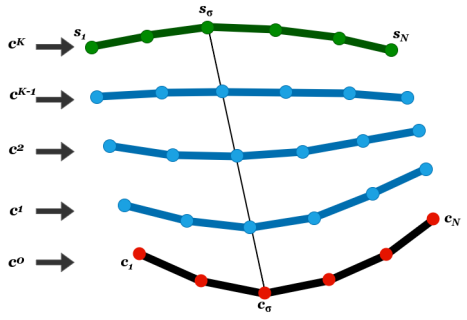
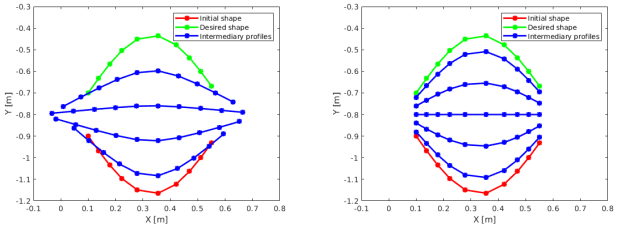


Figure 9: Illustration of the initial, desired, and intermediary shapes.



(a) The proposed ISG algorithm. (b) Algorithm proposed by Almaghout and Klimchik (2022a).

Figure 10: Comparison between the intermediary shapes generated by the ISG algorithm and the those generated by the linear interpolation based algorithm for the case of changing concavity.

Compared to the linear generation of the intermediary profiles methods proposed by Zhu et al. (2021) and

Algorithm 1: Intermediary Shapes Generation (ISG)

Given :
$\boldsymbol{c}^0$: DLO initial shape
$\boldsymbol{s}$: Desired shape
N: Number of DLO points
L: DLO length
λ: user-defined step

Begin :
$\boldsymbol{\delta} \leftarrow \boldsymbol{s} - \boldsymbol{c}^0$
$\sigma \leftarrow \arg\max(\boldsymbol{\delta}); \sigma \in \{1, ..., N\}$
$K \leftarrow \lfloor \frac{\delta_\sigma}{\lambda} \rfloor + 1$
`% Starting from` $\mathbf{c}_\sigma$ `and` $\mathbf{s}_\sigma$`, calculate angles between each two adjacent points`
for $n = \sigma$ *to* N **do**
  $\theta_{c_n} \leftarrow \text{atan2}\left((c_{ny} - c_{(n-1)y}), (c_{nx} - c_{(n-1)x})\right)$
  $\theta_{s_n} \leftarrow \text{atan2}\left((s_{ny} - s_{(n-1)y}), (s_{nx} - s_{(n-1)x})\right)$
end
for $n = \sigma$ *to* 1 **do**
  $\theta_{c_n} \leftarrow \text{atan2}\left((c_{ny} - c_{(n+1)y}), (c_{nx} - c_{(n+1)x})\right)$
  $\theta_{s_n} \leftarrow \text{atan2}\left((s_{ny} - s_{(n+1)y}), (s_{nx} - s_{(n+1)x})\right)$
end
`% Calculate the angular displacement for all points`
$\boldsymbol{\Theta} \leftarrow \Theta_n = \theta s_n - \theta c_n$
`% Calculate the angular step length`
$\vartheta_n^l \leftarrow \frac{\Theta_n}{K-1}$
`% Generate the intermediary shapes`
for $\kappa = 1$ *to* $K-1$ **do**
  $\boldsymbol{c}_\sigma^\kappa = \boldsymbol{c}_\sigma^{\kappa-1} + \lambda$
  for $n = \sigma + 1$ *to* N **do**
    $\boldsymbol{c}_n^\kappa \leftarrow \boldsymbol{c}_{n-1}^\kappa + \begin{bmatrix} l_s \cos(\theta c_{n-1} + \vartheta_n^a) \\ l_s \sin(\theta c_{n-1} + \vartheta_n^a) \end{bmatrix}$
  end
  for $n = \sigma - 1$ *to* 1 **do**
    $\boldsymbol{c}_n^\kappa \leftarrow \boldsymbol{c}_{n+1}^\kappa - \begin{bmatrix} l_s \cos(\theta c_{n+1} + \vartheta_n^a) \\ l_s \sin(\theta c_{n+1} + \vartheta_n^a) \end{bmatrix}$
  end
end
$\boldsymbol{c}^K = \boldsymbol{s}$

(Almaghout and Klimchik (2022a)), the ISG outputs more reasonable and feasible intermediary shapes, which make it applicable for large deformation scenarios. Figure 10 shows a comparison between the output of the proposed algorithm herein, Figure 10a and the one proposed earlier in (Almaghout and Klimchik (2022a)), Figure 10b. It can be seen that in Figure 10b, the generated intermediate profiles are not realistic. For instance, the third intermediary shape assumed the DLO to be fully stretched with length smaller than the DLO real length. Thus, if the robots end-effectors follow the intermediate points of the DLO ends, the DLO will be

compressed and an unpredictable behavior will occur, due to the increase of its internal potential energy. This undesirable behavior may lead to a failure in the manipulation task. On the other hand, in Figure 10a, the intermediary shapes generated by the ISG algorithm are realistic, and allow the DLO to be stretched to eliminate an excessive increase in potential energy and avoid the mentioned undesirable behaviors.

5. DLO Manipulation

Once we generate the intermediary shapes, the manipulators start guiding the DLO through these intermediary shapes towards the desired shape. To obtain the required displacement of the end-effectors configurations to manipulate the DLO accurately, we formulate manipulations task as an optimization control problem (OCP), where the DLO is guided sequentially through the intermediary shapes as follows:

$$\begin{aligned} &\text{for } \kappa = 1, \cdots, K: \\ &\min_{\boldsymbol{r}} \sum_{n=1}^{N} \left\| \boldsymbol{c}_n^{\kappa} - \boldsymbol{c}_n \right\| \\ &\qquad \text{such that: } \dot{\boldsymbol{c}} = \boldsymbol{J}\dot{\boldsymbol{r}} \\ &\qquad\qquad |\dot{\boldsymbol{r}}| \leq \boldsymbol{v} \end{aligned} \tag{13}$$

The first constraint represents an approximate model, mapping the motion to the DLO points and to the end-effectors. This model defines a Jacobian based on the diminishing rigidity property of the DLO, meaning that the rigidity of the motion of the DLO points decreases as the distance between the point and the end-effector increases. Berenson, in Berenson (2013), defines rigidity descent as exponentially proportional to the distance between end effector and grasped points. We use a Jacobian $\boldsymbol{J}_n^m$ to map each DLO point $\boldsymbol{c}_n$ to end-effector velocity $\dot{\boldsymbol{r}}_m$:

$$\dot{\boldsymbol{c}}_n = \boldsymbol{J}_n^m \dot{\boldsymbol{r}}_m \tag{14}$$

We compute $\boldsymbol{J}_n^m$ by assuming that the DLO is fully stretched and the point moves rigidly with the end-effector. Then, we multiply it by an exponential function of the distance:

$$\boldsymbol{J}_n^m = e^{-|D_n^m - d_n^m|} \begin{bmatrix} 1 & 0 & -\beta_m D_n^m \sin\theta_m \\ 0 & 1 & \beta_m D_n^m \cos\theta_m \end{bmatrix}. \tag{15}$$

In this equation, $D_n^m = \eta_n^m \cdot l_s$ (with η_n^m is the order of $\boldsymbol{c}_n$ with respect to $\boldsymbol{r}_m$) is the Euclidean distance between $\boldsymbol{c}_n$ and $\boldsymbol{r}_m$ when the DLO is fully stretched, $d_n^m = \left\| \boldsymbol{c}_n - \boldsymbol{r}_m \right\|$ is the Euclidean distance between $\boldsymbol{c}_n$ and $\boldsymbol{r}_m$ at any configuration of the DLO. Lastly, $\beta_1 = 1$, $\beta_2 = -1$, $\theta_1 = \varphi_1$, $\theta_2 = \pi - \varphi_2$.

In our previous work, (Almaghout and Klimchik (2022b)), we assumed that each DLO point is guided by two end-effectors. We observed that the impact of a small change of end-effector configuration on a DLO point is related to the distance between DLO point and end-effector. Thus in the addressed case, where the manipulation is quasi-static, the end-effector that is nearest to the DLO point has the major impact, while the impact on the farthest end-effector can be neglected. Thus, in contrast with that work, herein, we assign a one guiding end-effector for each DLO point: the end-effector guiding each DLO point $\boldsymbol{c}_n$ is the one nearest to $\boldsymbol{c}_n$, so the guiding end effector index is:

$$\arg min\left\{ D_n^1, D_n^2 \right\} = \{1, 2\}. \tag{16}$$

Hence, for a DLO of N points, where N is even, the Jacobian is obtained as follows:

$$\dot{\boldsymbol{c}} = \begin{bmatrix} \boldsymbol{J}_{N\times 3}^1 & \boldsymbol{0}_{N\times 3} \\ \boldsymbol{0}_{N\times 3} & \boldsymbol{J}_{N\times 3}^2 \end{bmatrix} \dot{\boldsymbol{r}} \tag{17}$$

The second constraint in (13) imposes bounds on the linear and angular velocities of the robots' end-effectors: $|\dot{\boldsymbol{r}}|$ is the element-wise absolute values vector of $\dot{\boldsymbol{r}}$; and $\boldsymbol{v}$ is a 6 vector of the maximum linear and angular robot velocities.

To evaluate our approach, we define the error as the deflection between the DLO points position and the desired one, as follows:

$$\boldsymbol{e} = \begin{bmatrix} e_1 & \dots & e_n & \dots & e_N \end{bmatrix}^T \tag{18}$$

where

$$e_n = \left| \boldsymbol{c}_n^K - \boldsymbol{c}_n \right| \tag{19}$$

As performance metrics, we consider the mean and standard deviation of the error defined in (18):

- the mean error:

$$e_{mean} = \frac{1}{N} \sum_{n=1}^{N} e_n, \tag{20}$$

- the standard deviation of the error:

$$e_{std} = \sqrt{\frac{\sum_{n=1}^{N} \left(e_n - e_{mean} \right)}{N-1}}. \tag{21}$$

We will use these metrics further to evaluate the approach in simulations and in real-life experiments. Additionally, the mean error e_{mean} will be considered as system convergence criterion.

6. Simulation Study

To evaluate the proposed algorithm in simulation, we first model the DLO as a mass-spring model. Then, we realize several simulations, to study the performance of our approach, considering several factors. The simulations are carried out in MATLAB2020b running on Ubuntu 18.04.6 operating system on a computer equipped with Intel Core i7-10510U CPU running at 1.8 GHz and 16 GB of RAM.

6.1. DLO Simulation

We use the mass-spring model for evaluation purpose since it is an intuitive practical physical model, easy to implement, and computationally efficient(Lv et al. (2020)). Due to these advantages, this model is widely used for simulation of deformable objects, such as cables Lv et al. (2017)), clothes (ElBadrawy and Hemayed (2011)), and cloak (Patete, Iacono, Spadea, Trecate, Vergnaghi, Mainardi and Baroni (2013)). The model discretizes the DLO into N mass points. The position of these mass points represents the shape of the DLO. The mass points are connected by $N-1$ massless stiff linear springs to preserve the length of the DLO, while representing its stretchabilty. The model has $N-2$ torsion springs situated at the mass points, excluding the end points, to describe the elastic bending behaviour. Figure 11 depicts the DLO model, including the springs.

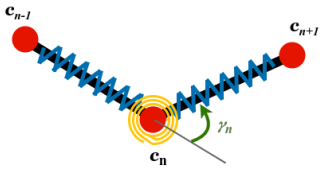


Figure 11: Mass-spring model used to simulate a cable.

For a displacement of the DLO ends, the positions of the internal DLO points are updated, based on the stretching and bending stiffness of the DLO. We introduce an energy-based DLO model where the object energies are defined as follows: the stretching energy is

$$E_s = \sum_{n=2}^{N} \frac{k_s}{2} \left(\|\boldsymbol{c}_n - \boldsymbol{c}_{n-1}\| - l_s \right)^2, \tag{22}$$

and the bending energy is

$$E_b = \sum_{n=2}^{N-1} \frac{k_b}{2} \gamma_n^2. \tag{23}$$

In the above equations, k_s and k_b are the stiffness parameters of the linear and torsional springs, respectively; γ_n is the bending angle of $\boldsymbol{c}_n$, shown in Figure 11, and given by:

$$\gamma_n = \tan^{-1} \left(\frac{\left\| \left(\boldsymbol{c}_{n+1} - \boldsymbol{c}_n\right) \times \left(\boldsymbol{c}_n - \boldsymbol{c}_{n-1}\right) \right\|}{\left(\boldsymbol{c}_{n+1} - \boldsymbol{c}_n\right)^T \left(\boldsymbol{c}_n - \boldsymbol{c}_{n-1}\right)} \right). \tag{24}$$

Then, for any update in the DLO ends position, the inner points $\boldsymbol{c}_{inner}$ position are updated to minimize the total energy

$$\min_{\boldsymbol{c}_{inner}} \{E_s + E_b\}. \tag{25}$$

Model (25) is used further to evaluate the proposed approach in simulations.

Table 5
DLOs geometric and elastic parameters

Model	L $[mm]$	d $[mm]$	E $[MPa]$
$DLO1$	500	4.0	100
$DLO2$	700	7.0	126

We use two DLO models, with parameters presented in Table 5. Their stiffness coefficients are computed based on the formulas proposed in Lv et al. (2017):

- the linear stiffness coefficient: $k_s = E \cdot A / l_s$,
- the bending stiffness coefficient: $k_b = 3 \cdot E \cdot I / l_s$.

In the above equations: E is Young's modulus, A and I are the DLO's cross sectional area and moment of inertia, respectively. For circular cross-section DLO (e.g., cables) of diameter d: $A = \pi \cdot d^2/4$ and $I = \pi \cdot d^4/64$.

Then, for any update in the DLO ends, the inner points $\boldsymbol{p}_{inner}$ positions are updated to minimize the total energy

$$\min_{\boldsymbol{p}_{inner}} \boldsymbol{E} \tag{26}$$

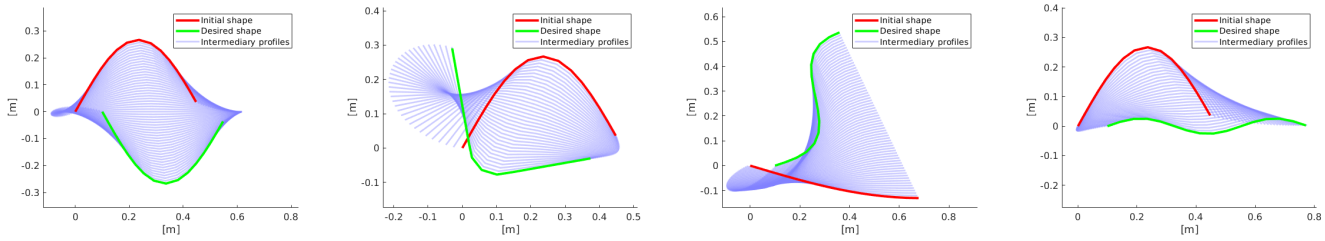
Thus, for each update in the $\boldsymbol{p}_{control}$ position, the model updates its $\boldsymbol{p}_{inner}$ position, to minimize the internal energies and to put the DLO in equilibrium configuration. Further, we will use this model for the simulation of DLO manipulation, where the control will guide the $\boldsymbol{p}_{control}$ to deform the DLO into the desired shape.

6.2. Study for different N values

The approach discretizes both the DLO and the desired shape into N points. Choosing the proper number of DLO points N plays a crucial rule. Reducing the number of points will not provide a realistic approximation, and will negatively impact final shape accuracy, although it reduces the computational cost, and vice versa. We investigate the impact of N on the final result, by a comparative study. The study considers DLO models with $N = 8,\ 10,\ 12$, and 14. It is worth mentioning that we tested the DLO models for the number of points from 4 to 20. For $N = 4$ and 6 the model was unrealistic and for $N > 14$ the model is redundant and the computational cost is very high. Four desired shapes are considered, **U**, **L**, **S**, and **M**, shown in Figure 12. These scenarios represents different levels of complexity, with different inflectional points. Our study takes into account the following metrics:

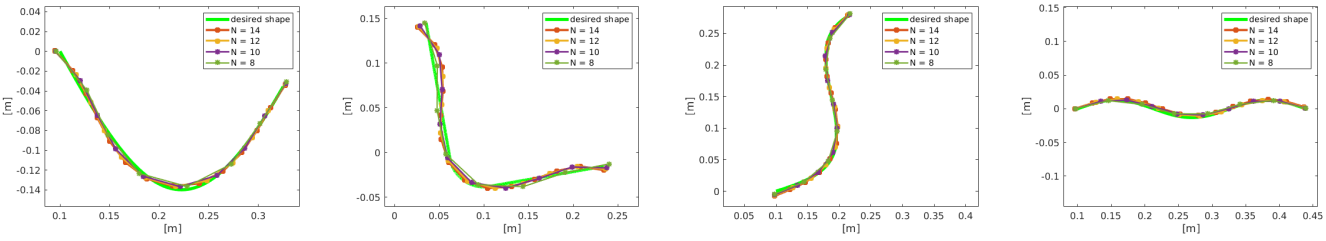
- the error mean defined by (20).
- the error standard deviation defined by (21).
- the number of iterations.

The convergence criterion of the optimization control problem is $e_{mean} < 0.01L$, the maximum number of iterations is



(a) U-shape (b) L-shape (c) S-shape (d) M-shape

Figure 12: The desired shapes of the considered scenarios U, L, S, and M; and the intermediary shapes generated by ISG algorithm from the initial shapes towards the desired ones.



(a) U-shape (b) L-shape (c) S-shape (d) M-shape

Figure 13: Final shapes of $DLO1$ ($L = 700mm, d = 7mm, E = 126MPa$), for different number of feature points $N = 8,\ 10,\ 12,\ 14$.

$20 \cdot K$, where K is total number of intermediary and desired shapes defined in (3), and $\lambda = 10mm$.

The intermediary shapes generated by the ISG algorithm are depicted in Figure 12. It can be noticed that the ISG algorithm successfully outputs realistic shapes, despite the deformation complexity. It is also suitable for the opposite concavity scenario, as shown in figure 12a, where the DLO is fully stretched and then bent towards the desired shape.

Figures 13 and 14 show the final shapes of $DLO1$ and $DLO2$, respectively, at the end of the manipulation task. It can be seen that the developed approach achieved the task and fitted the DLO models to the desired shapes for all N values. We show the trials' mean error and standard deviation for $DLO1$ and $DLO2$ in Fig. 15 and 16, respectively. One can observe that although all trials succeeded, 6 out of 32 did not reach the convergence threshold, one trial for $N = 10$, two for $N = 12$, and three for $N = 14$. the figures presents an overall tendency of the error to increase as N increases. The relatively small standard deviation e_{std} shows that the points individual error have dispersion about the mean error, as is also confirmed by Figures 13 and 14. Figure 17 shows the required number of iterations and the execution time of each trial. It illustrates that both the number of iterations and execution time increase as N increases. Thus, taking into account these results, the proposed approach performs well regardless the N value. However, N significantly impacts the realisticness of representing the DLO and the execution time required to accomplish the task.

6.3. Different Initial Conditions

The objective of the next study is to assess the performance of the approach under different initial conditions (shapes) while targeting the same desired shape. For this test, we utilized the $DLO1$ method with parameters $N = 10$, $\lambda = 10$mm, and a maximum of $20 \cdot K$ iterations.

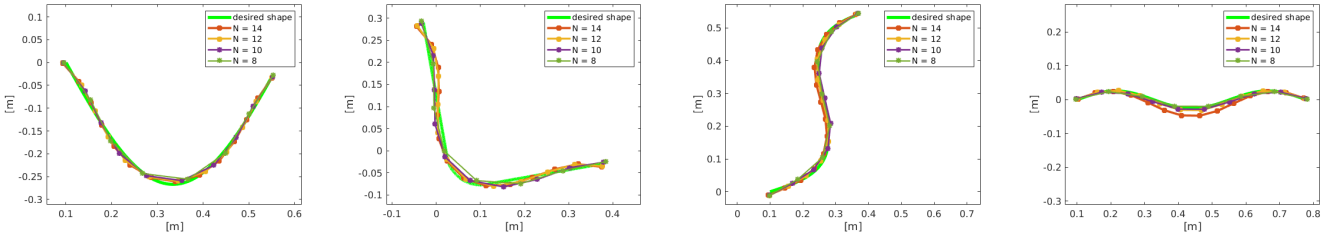
In Figures 18, 19, 20, and 21, we present five distinct initial shapes, corresponding to the desired shapes **U**, **L**, **S**, and **M**, respectively. It is evident that despite variations in the complexity of the initial shapes, the approach successfully achieved the desired shapes for all cases.

The e_{mean} error for each desired shape roughly varies depending on the initial shape. For the U-shape, the overall e_{mean} error was 4.40mm, with a narrow standard deviation of 0.54mm. For the L-shape, it was 7.24mm, with a standard deviation of 0.81mm. Meanwhile, the S-shape had a mean error of 3.84mm, with a remarkably low standard deviation of 0.30mm. Lastly, the M-shape achieved an average e_{mean} error of 3.26mm, with a standard deviation of 1.22mm. For a comprehensive overview, Table 6 summarizes the e_{mean} for each desired shape along with the corresponding initial shapes.

These findings provide valuable insights into the robustness and adaptability of our approach to diverse initial conditions, ultimately yielding the desired shapes with remarkable accuracy.

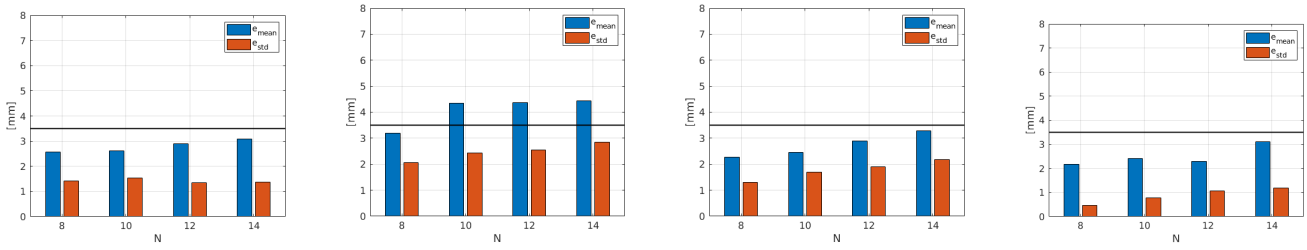
6.4. The impact of λ

To determine the suitable step value λ for generating local intermediary shapes, we conducted a study examining the impact of λ on the performance, across eight different



(a) U-shape (b) L-shape (c) S-shape (d) M-shape

Figure 14: Final shapes of $DLO2$ ($L = 500mm, d = 4mm, E = 100MPa$), for different number of feature points $N = 8, 10, 12, 14$.



(a) U-shape (b) L-shape (c) S-shape (d) M-shape

Figure 15: The approach efficiency represented by mean error e_{mean} and standard deviation e_{std} in simulation experiments of $DLO1$ ($L = 700mm, d = 7mm, E = 126MPa$).

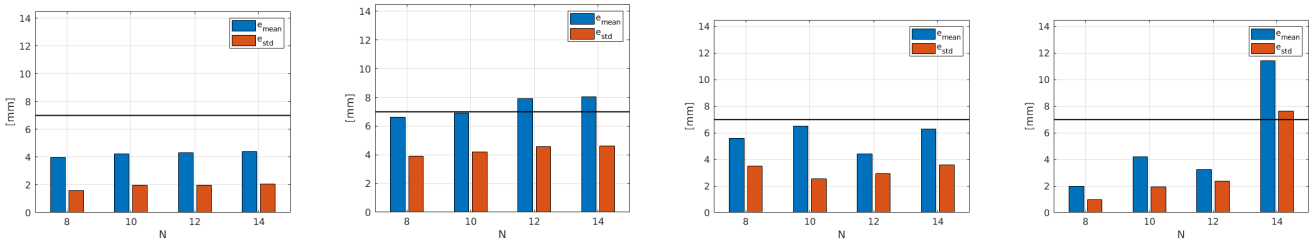
Table 6
Mean DLO shape error e_{mean} $[mm]$ in the simulation experiments for different initial shapes.

Initial Shapes	***1***	***2***	***3***	***4***	***5***
U-shape	4.05	4.02	4.43	5.34	4.18
L-shape	6.55	6.64	6.78	8.07	8.20
S-shape	3.79	4.22	4.11	3.56	3.56
M-shape	2.75	4.59	4.08	3.46	1.45

scenarios: U1: Figure 18a, U2: Figure 18b, L1: Figure 19a, L2: Figure 19b, S1: Figure 20a, S2: Figure 20b, M1: Figure 21a, and M2: Figure 21b. These scenarios encompass various desired shapes and corresponding initial shapes, drawing from the simulation study in Section 6.3.

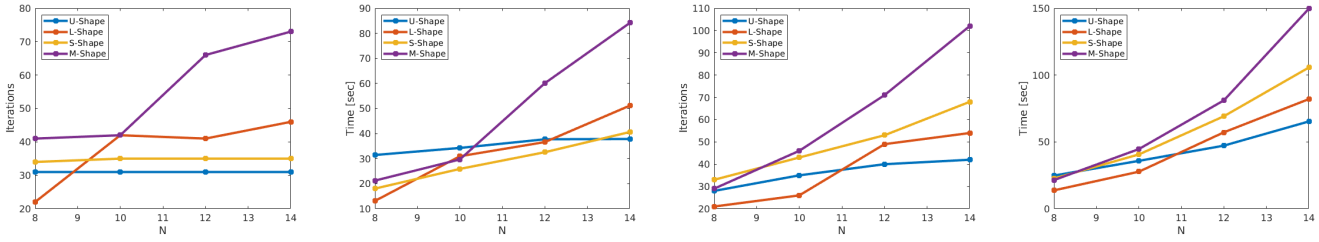
We explored the variation of λ within the range of $1m$ to $100mm$ for a step of $1mm$. The results are depicted in Figures 22 and 23, showing the relationship between λ and the final e_{mean} (mean error) as well as the number of iterations required, respectively. It is notable that, as λ increases, the number of iterations decreases. For $\lambda < 10mm$, the number of iterations for trial $M1$ was exceptionally high, exceeding 1400. On the other hand, the final e_{mean} remained relatively steady for all λ values below $50mm$. However, beyond $\lambda = 50mm$, the system began to encounter challenges, and the e_{mean} increased, indicating unsuccessful trials.

Based on this intensive study, we conclude that the system performs effectively for λ values between 10 and



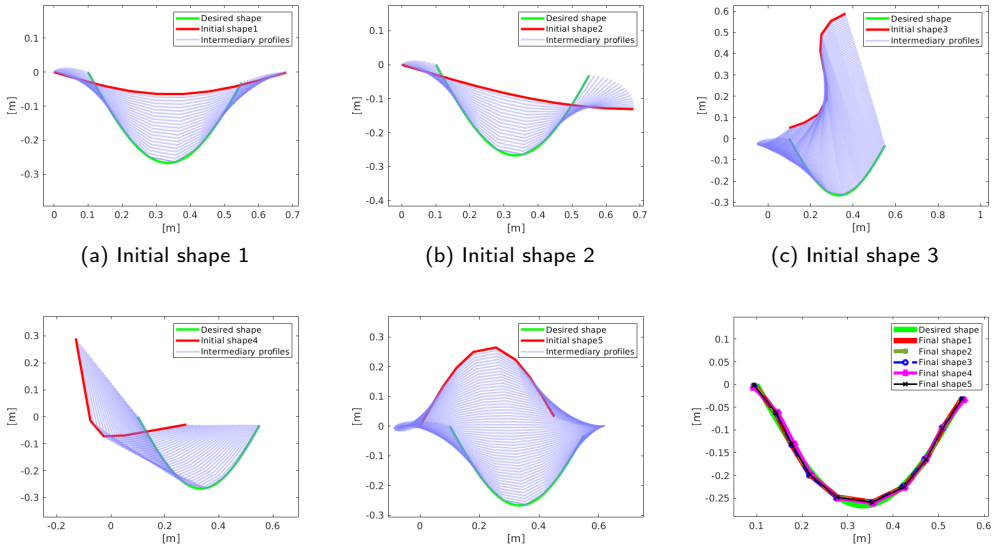
(a) U-shape (b) L-shape (c) S-shape (d) M-shape

Figure 16: The approach efficiency represented by mean error e_{mean} and standard deviation e_{std} in simulation experiments of $DLO2$ ($L = 500mm, d = 4mm, E = 100MPa$)



(a) $DLO1$: iterations (b) $DLO1$: execution time (c) $DLO2$: iterations (d) $DLO2$: execution time

Figure 17: The relation between iterations and execution time and number of DLO points N in experiments of $DLO1$ ($L = 700mm, d = 7mm, E = 126MPa$) and $DLO2$ ($L = 500mm, d = 4mm, E = 100MPa$).



(a) Initial shape 1 (b) Initial shape 2 (c) Initial shape 3

(d) Initial shape 4 (e) Initial shape 5 (f) Final shapes

Figure 18: The considered initial shapes (a) to (e) alongside the ISG algorithm output (intermediary shapes) and final shapes (f) for U-shape scenario.

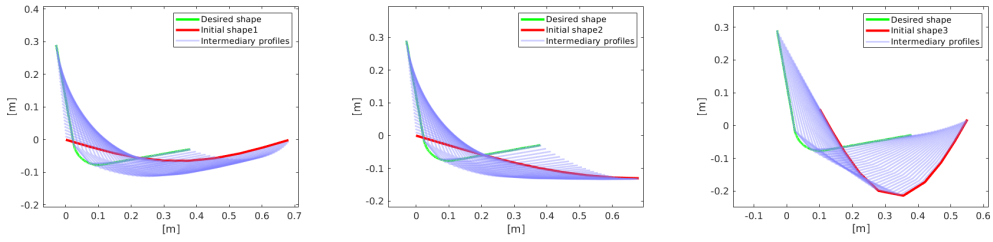
$50mm$. The aforementioned findings present crucial perspectives, which guide us towards the selection of an optimal step value, to attain precise shape manipulation through our approach.

7. Experiments

7.1. Hardware Setup

Figure 24 shows the experimental setup used to validate our theoretical findings. Two KUKA LBR iiwa manipulators manipulate a DLO of length $L = 350mm$, and diameter $d = 9mm$. Each manipulator is equipped with a 3D printed tool designed to rigidly hold the DLO's end. We utilize an Intel Realsense D435 camera for top-view vision. After calibration, we use the VFPs algorithm proposed in our previous work (Almaghout and Klimchik (2022b)) to detect and track the DLO, by sampling it into N feature points and tracking their position. Having one of the tools marked, the VFPs algorithm detects the marked tool as the starting point, via color-based segmentation. Next, the DLO is detected and segmented by a sequence of edge-detection methods and morphological operations. Then, we obtain the center line of the DLO, with a one-pixel width by applying a thinning algorithm Guo and Hall (1992). The algorithm starts sampling by sliding a circular mask, whose center is placed at the starting point. We consider the intersection between the DLO centerline and the mask as a VFP. The mask continues sliding to the obtained VFP, the new intersection point is assigned as the new VFP, and so forth until the other end of the DLO. Finally, the algorithm returns the DLO sampled as N virtual feature points. Figure 25 shows a diagram of the VFPs algorithm, as well as its input and output frames.

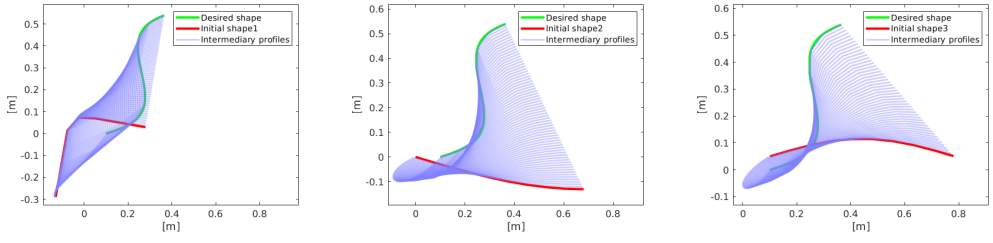
We built the algorithm using the Robotic Operating System (ROS, Stanford Artificial Intelligence Laboratory et al.), and wrote the optimization control problem with CasADi (Andersson, Gillis, Horn, Rawlings and Diehl (2019)). The robots linear and angular velocities are limited to $0.030m/s$ and $0.080rad/s$, respectively. The convergence threshold is $e_{mean} = 10mm$; we set $\lambda = 10mm$ and the maximum number of iterations to $20 \cdot K$.



(a) Initial shape 1 (b) Initial shape 2 (c) Initial shape 3

(d) Initial shape 4 (e) Initial shape 5 (f) Final shapes

Figure 19: The considered initial shapes (a) to (e) alongside the ISG algorithm output (intermediary shapes) and final shapes (f) for L-shape scenario.



(a) Initial shape 1 (b) Initial shape 2 (c) Initial shape 3

(d) Initial shape 4 (e) Initial shape 5 (f) Final shapes

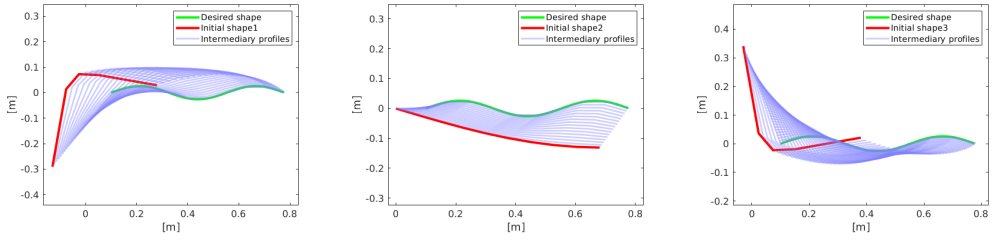
Figure 20: The considered initial shapes (a) to (e) alongside the ISG algorithm output (intermediary shapes) and final shapes (f) for S-shape scenario.

We evaluate the approach through two sets of experiments. In the first set, the robots manage to deform the DLO to the U, L, S, M shapes, seen in the simulations. In the second set, we address opposite concavity scenarios, where the desired shapes are concave, while the initial shapes are convex (or vice versa).

7.2. U-L-S-M Shapes

We first tested the approach on the same desired shapes considered in the simulations shown in Fig. 12. Figure 26 shows the initial, intermediary, and final shapes **U**, **L**, **S**, and **M**. The figure shows how the VFPs algorithm detects and samples the DLO, the intermediary shapes generated by the ISG algorithm, and the final results of the manipulation for all shapes. It can be noticed that the system properly deformed the DLO to the desired shape.

The mean error gradually decreased during the manipulation process for all experiments, see Fig. 27. The robots successfully accomplished all tasks, with a maximum e_{mean}



(a) Initial shape 1 (b) Initial shape 2 (c) Initial shape 3

(d) Initial shape 4 (e) Initial shape 5 (f) Final shapes

Figure 21: The considered initial shapes (a) to (e) alongside the ISG algorithm output (intermediary shapes) and final shapes (f) for M-shape scenario.

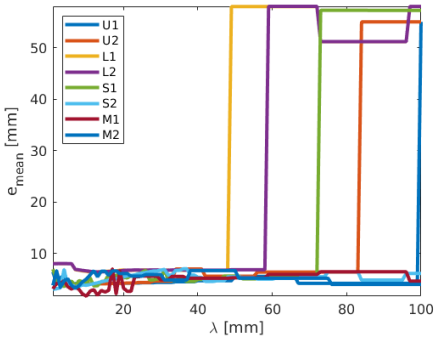


Figure 22: The relation between the mean shape error e_{mean} and the user-defined step value λ in the different considered case studies: U1: Figure 18a, U2: Figure 18b, L1: Figure 19a, L2: Figure 19b, S1: Figure 20a, S2: Figure 20b, M1: Figure 21a, and M2: Figure 21b

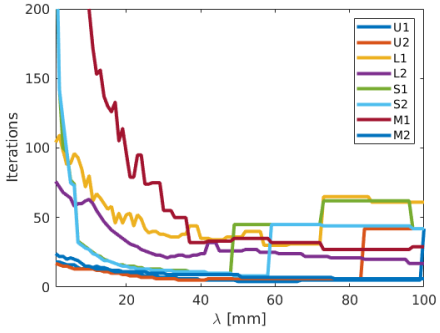


Figure 23: The relation between the number of iterations executed and the user-defined step value λ in the different considered case studies: U1: Figure 18a, U2: Figure 18b, L1: Figure 19a, L2: Figure 19b, S1: Figure 20a, S2: Figure 20b, M1: Figure 21a, and M2: Figure 21b

of 11.7 mm in the most challenging case and a minimum of $e_{mean} = 5.2$ mm in the best-case scenario.

Table 7 provides a comparison of e_{mean} and e_{std} (standard deviation) between real-life experiments and simulations, for $N = 10$. These results prove the effectiveness and reliability of our approach in achieving accurate shape manipulation, in both simulations and real-life experiments.

7.3. Opposite-concavity Shapes

We proceeded to explore more challenging scenarios, where the initial shapes had opposite concavities compared to the desired shapes. It is worth mentioning that, this challenge has never been addressed in the previous works on manipulation of a DLO grasped by its two ends. Previous works investigated local deformations, when the desired

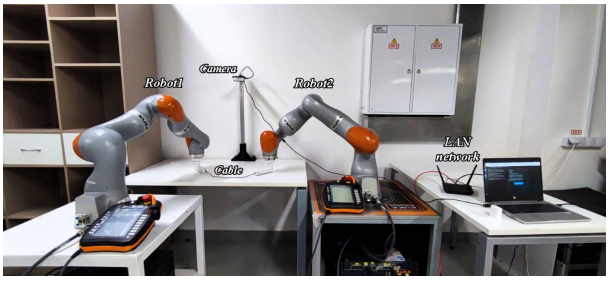


Figure 24: The hardware setup for the validation in real-life experiments.

shape had no more than two inflectional points and when the initial shape is *locally* close to the desired one (Zhu et al. (2018), Yu et al. (2022), Huang et al. (2023)). Figure

Table 7
Comparison between simulations and experiments based on mean shape error e_{mean} and on the error standard deviation e_{std}.

	U-shape		*L-shape*		*S-shape*		*M-shape*	
	sim.	exp.	sim.	exp.	sim.	exp.	sim.	exp.
e_{mean} $[mm]$	4.0	11.7	6.9	10.9	5.2	10.6	4.9	5.2
e_{std} $[mm]$	2.0	5.7	4.2	5.0	2.4	6.4	3.0	3.0

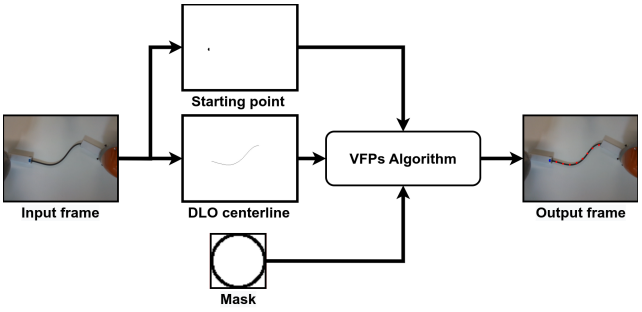


Figure 25: The VFPs algorithm. The image of the DLO centerline, starting point, and mask are in inverted color.

Table 8
The approach efficiency in the opposite-concavity experiments represented by the mean shape error e_{mean} and the error standard deviation e_{std}.

Scenario	*1*	*2*	*3*	*4*
$e_{mean}[mm]$	12.4	9.0	8.1	4.5
$e_{std}[mm]$	5.1	3.7	3.1	3.3

28 presents four of these scenarios, showing the shapes at their initial and final shapes. Notably, the ISG algorithm effectively generated intermediary shapes that facilitated reversing the DLO concavity, preventing it from staying trapped in its initial concavity. The final results show that the robots successfully deformed the DLOs, to precisely fit the desired shapes.

Figure 29 presents the reduction in error as the robots guided the DLO towards its desired shapes. In all of these scenarios, the system effectively managed to finish the task with a maximum mean error below $12.5mm$. Table 8 shows the mean and standard deviation of the errors observed in these scenarios.

These findings emphasize the effectiveness and adaptability of our approach, in handling challenging situations involving opposite concavities. The approach showcases promising performance in achieving accurate shape manipulation, even in complex scenarios with varying curvatures.

8. Conclusions

This article proposes a new methodology to achieve the shape control of DLO in 2D workspace, by two manipulators working collaboratively. The DLO is grasped at its ends by two manipulators. We introduce the ISG algorithm to generate a set of intermediary shapes. The control problem is defined as an optimization problem. We carry out an intensive evaluation, to test the approach performance in simulation and real-life experiments, in a variety of scenarios at different levels of complexity. For the first time, the challenge of opposite concavity scenarios, is addressed successfully by our approach. The robots realize all desired shapes with a high final accuracy. The experiments prove the robustness and capability of our approach in effectively performing complex deformation, which is the main contribution of our work.

Acknowledgement

This work was Supported by Russian Scientific Foundation (Project number 22-41-02006).

References

Almaghout, K., Boby, R.A., Othman, M., Shaarawy, A., Klimchik, A., 2021. Robotic pick and assembly using deep learning and hybrid vision/force control, in: 2021 International Conference" Nonlinearity, Information and Robotics"(NIR), IEEE. pp. 1–6.

Almaghout, K., Klimchik, A., 2022a. Planar shape control of deformable linear objects. IFAC-PapersOnLine 55, 2469–2474.

Almaghout, K., Klimchik, A., 2022b. Vision-based robotic comanipulation for deforming cables. Russian Journal of Nonlinear Dynamics 18, 843–858.

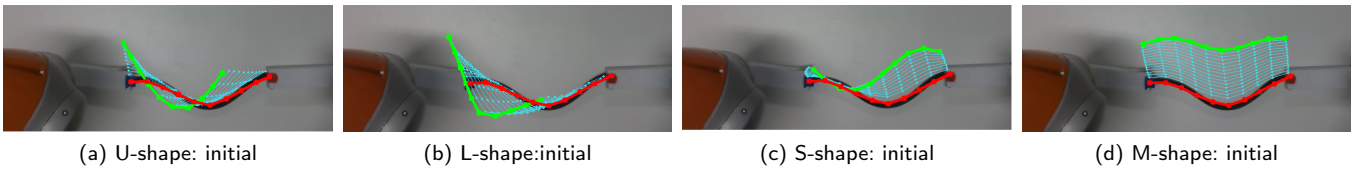
Andersson, J.A.E., Gillis, J., Horn, G., Rawlings, J.B., Diehl, M., 2019. CasADi – A software framework for nonlinear optimization and optimal control. Mathematical Programming Computation 11, 1–36. doi:10.1007/s12532-018-0139-4.

Berenson, D., 2013. Manipulation of deformable objects without modeling and simulating deformation, in: 2013 IEEE/RSJ International Conference on Intelligent Robots and Systems, IEEE. pp. 4525–4532.

Delgado, A., Corrales, J.A., Mezouar, Y., Lequievre, L., Jara, C., Torres, F., 2017. Tactile control based on gaussian images and its application in bimanual manipulation of deformable objects. Robotics and Autonomous systems 94, 148–161.

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

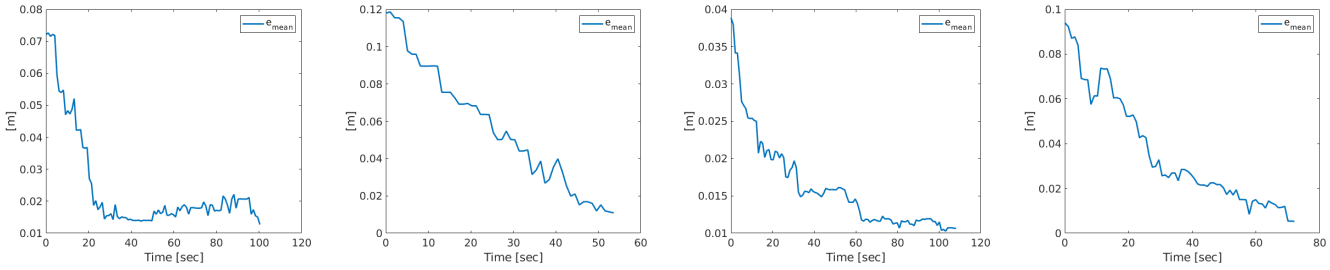
ElBadrawy, A.A., Hemayed, E.E., 2011. Speeding up cloth simulation by linearizing the bending function of the physical mass-spring model, in: 2011 International Conference on 3D Imaging, Modeling, Processing, Visualization and Transmission, IEEE. pp. 101–107.



(a) U-shape: initial (b) L-shape:initial (c) S-shape: initial (d) M-shape: initial

(e) U-shape: final (f) L-shape: final (g) S-shape: final (h) M-shape: final

Figure 26: U-L-S-M experiment scenarios. Figures (a) to (d) show the initial DLO shape (red) along with the intermediary shapes (blue), and the desired shape (green). Figures (e) to (h) show the final results.



(a) U-shape (b) L-shape (c) S-shape (d) M-shape

Figure 27: The evolution of mean shape error e_{mean} in the U-L-S-M shapes experiment scenarios. The line graphs demonstrate the gradual decline in the curves of the mean shape error during the manipulation.

Guo, Z., Hall, R.W., 1992. Fast fully parallel thinning algorithms. CVGIP: image understanding 55, 317–328.

Heisler, P., Steinmetz, P., Yoo, I.S., Franke, J., 2017. Automatization of the cable-routing-process within the automated production of wiring systems, in: Applied Mechanics and Materials, Trans Tech Publ. pp. 186–192.

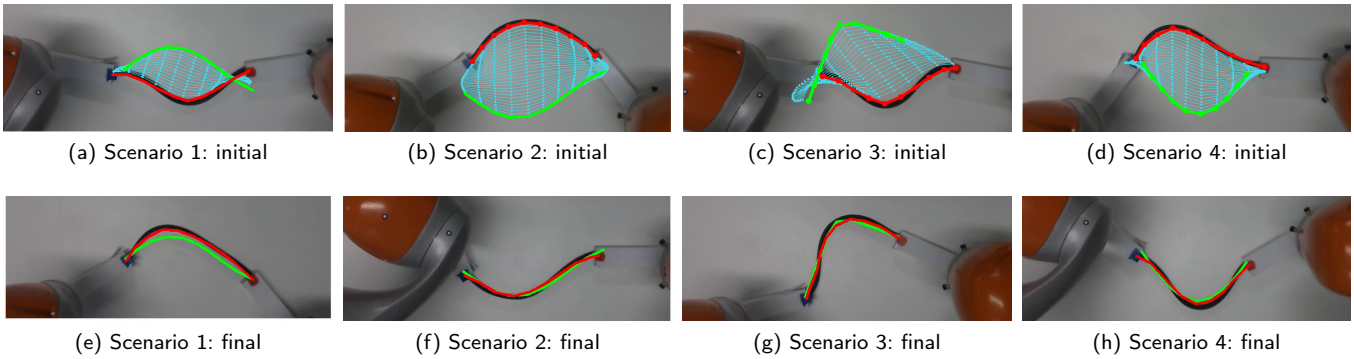
Huang, Y., Xia, C., Wang, X., Liang, B., 2023. Learning graph dynamics with external contact for deformable linear objects shape control. IEEE Robotics and Automation Letters .

Jin, S., Wang, C., Tomizuka, M., 2019. Robust deformation model approximation for robotic cable manipulation, in: 2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), IEEE. pp. 6586–6593.

Koessler, A., Filella, N.R., Bouzgarrou, B.C., Lequièvre, L., Ramon, J.A.C., 2021. 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. pp. 1637–1643.

Laezza, R., Gieselmann, R., Pokorny, F., Karayiannidis, Y., 2021. Shape control of elastoplastic deformable linear objects through reinforcement learning, in: Proc. of the IEEE International Conference on Robotics and Automation.

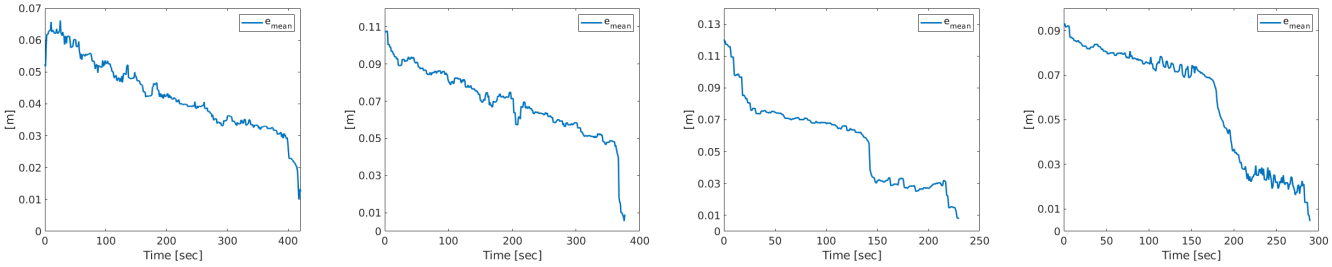
Laezza, R., Karayiannidis, Y., 2021. Learning shape control of elastoplastic deformable linear objects, in: 2021 IEEE International Conference on



(a) Scenario 1: initial (b) Scenario 2: initial (c) Scenario 3: initial (d) Scenario 4: initial

(e) Scenario 1: final (f) Scenario 2: final (g) Scenario 3: final (h) Scenario 4: final

Figure 28: Opposite concavity experiment scenarios. experiment scenarios. Figures (a) to (d) show the initial DLO shape (red) along with the intermediary shapes (blue), and the desired shape (green). Figures (e) to (h) show the final results.



(a) Scenario 1 (b) Scenario 2 (c) Scenario 3 (d) Scenario 4

Figure 29: The evolution of mean shape error e_{mean} in the opposite concavity experiment scenarios. The line graphs demonstrate the gradual decline in the curves of the mean shape error during the manipulation.

Robotics and Automation (ICRA), IEEE. pp. 4438–4444.

Lagneau, R., Krupa, A., Marchal, M., 2020. Automatic shape control of deformable wires based on model-free visual servoing. IEEE Robotics and Automation Letters 5, 5252–5259.

Li, X., Wang, Z., Liu, Y.H., 2019. Sequential robotic manipulation for active shape control of deformable linear objects, in: 2019 IEEE International Conference on Real-time Computing and Robotics (RCAR), IEEE. pp. 840–845.

Linn, J., Dreßler, K., 2017. Discrete cosserat rod models based on the difference geometry of framed curves for interactive simulation of flexible cables. Math for the Digital Factory , 289–319.

Liu, F., Su, E., Lu, J., Li, M., Yip, M.C., 2022. Differentiable robotic manipulation of deformable rope-like objects using compliant position-based dynamics. arXiv preprint arXiv:2202.09714 .

Liu, F., Su, E., Lu, J., Li, M., Yip, M.C., 2023. Robotic manipulation of deformable rope-like objects using differentiable compliant position-based dynamics. IEEE Robotics and Automation Letters .

Lv, N., Liu, J., Ding, X., Liu, J., Lin, H., Ma, J., 2017. Physically based real-time interactive assembly simulation of cable harness. Journal of Manufacturing Systems 43, 385–399.

Lv, N., Liu, J., Xia, H., Ma, J., Yang, X., 2020. A review of techniques for modeling flexible cables. Computer-Aided Design 122, 102826.

Nair, A., Chen, D., Agrawal, P., Isola, P., Abbeel, P., Malik, J., Levine, S., 2017. Combining self-supervised learning and imitation for vision-based rope manipulation, in: 2017 IEEE international conference on robotics and automation (ICRA), IEEE. pp. 2146–2153.

Patete, P., Iacono, M.I., Spadea, M.F., Trecate, G., Vergnaghi, D., Mainardi, L.T., Baroni, G., 2013. A multi-tissue mass-spring model for computer assisted breast surgery. Medical engineering & physics 35, 47–53.

Qi, J., Ma, G., Zhou, P., Zhang, H., Lyu, Y., Navarro-Alarcon, D., 2022. Towards latent space based manipulation of elastic rods using autoencoder models and robust centerline extractions. Advanced Robotics 36, 101–115.

Ruan, M., McConachie, D., Berenson, D., 2018. 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), IEEE. pp. 512–519.

Sanchez, J., Corrales, J.A., Bouzgarrou, B.C., Mezouar, Y., 2018. Robotic manipulation and sensing of deformable objects in domestic and industrial applications: a survey. The International Journal of Robotics Research 37, 688–716.

Sanchez, J., Mohy El Dine, K., Corrales, J.A., Bouzgarrou, B.C., Mezouar, Y., 2020. Blind manipulation of deformable objects based on force sensing and finite element modeling. Frontiers in Robotics and AI 7, 73.

Servin, M., Lacoursiere, C., 2008. Rigid body cable for virtual environments. IEEE Transactions on Visualization and Computer Graphics 14, 783–796.

Stanford Artificial Intelligence Laboratory et al., . Robotic operating system. URL: https://www.ros.org.

Tang, T., Wang, C., Tomizuka, M., 2018. A framework for manipulating deformable linear objects by coherent point drift. IEEE Robotics and Automation Letters 3, 3426–3433.

Trommnau, J., Kühnle, J., Siegert, J., Inderka, R., Bauernhansl, T., 2019. Overview of the state of the art in the production process of automotive wire harnesses, current research and future trends. Procedia CIRP 81, 387–392.

Valentini, P.P., Pennestrì, E., 2011. Modeling elastic beams using dynamic splines. Multibody system dynamics 25, 271–284.

Wang, C., Zhang, Y., Zhang, X., Wu, Z., Zhu, X., Jin, S., Tang, T., Tomizuka, M., 2022. Offline-online learning of deformation model for cable manipulation with graph neural networks. IEEE Robotics and Automation Letters 7, 5544–5551.

Xu, L., Liu, Q., 2018. Real-time inextensible surgical thread simulation. International Journal of Computer Assisted Radiology and Surgery 13, 1019–1035.

Yan, M., Zhu, Y., Jin, N., Bohg, J., 2020. Self-supervised learning of state estimation for manipulating deformable linear objects. IEEE robotics and automation letters 5, 2372–2379.

Yan, W., Vangipuram, A., Abbeel, P., Pinto, L., 2021. Learning predictive representations for deformable objects using contrastive estimation, in: Conference on Robot Learning, PMLR. pp. 564–574.

Yin, H., Varava, A., Kragic, D., 2021. Modeling, learning, perception, and control methods for deformable object manipulation. Science Robotics 6, eabd8803.

Yu, M., Zhong, H., Li, X., 2022. Shape control of deformable linear objects with offline and online learning of local linear deformation models, in: 2022 International Conference on Robotics and Automation (ICRA), IEEE. pp. 1337–1343.

Zakaria, M.H.D., Aranda, M., Lequièvre, L., Lengagne, S., Ramón, J.A.C., Mezouar, Y., 2022. Robotic control of the deformation of soft linear objects using deep reinforcement learning, in: 2022 IEEE 18th International Conference on Automation Science and Engineering (CASE), IEEE. pp. 1516–1522.

Zhang, W., Schmeckpeper, K., Chaudhari, P., Daniilidis, K., 2021. Deformable linear object prediction using locally linear latent dynamics, in: 2021 IEEE International Conference on Robotics and Automation (ICRA), IEEE. pp. 13503–13509.

Zhu, J., Cherubini, A., Dune, C., Navarro-Alarcon, D., Alambeigi, F., Berenson, D., Ficuciello, F., Harada, K., Kober, J., Li, X., et al., 2022. Challenges and outlook in robotic manipulation of deformable objects. IEEE Robotics & Automation Magazine 29, 67–77.

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

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