Deformable Linear Objects Manipulation with Online Model Parameters Estimation

Alessio Caporali, Piotr Kicki, Kevin Galassi, Riccardo Zanella, Krzysztof Walas and Gianluca Palli

***Abstract*—This paper tackles the robotic manipulation of Deformable Linear Objects (DLOs), focusing on the shape control task. A neural network (NN) is trained to replicate the DLO dynamics using data generated by an analytical model. Then, the NN-based model is utilized for shape control, where manipulation actions are optimized via gradient descent. Simultaneously, the same NN undergoes gradient-based optimization to adjust several model parameters given the observed real-world DLO dynamics. Experimental validation showcases the effectiveness and efficiency of the proposed approach across diverse DLOs, surfaces, and target shapes.**

***Index Terms*—deformable linear objects, shape control**

I. Introduction

Robotic handling of Deformable Linear Objects (DLOs) like ropes, cables, and wiring harnesses poses significant challenges both from the perception and manipulation perspectives. Indeed, it is difficult to detect DLOs [1], [2] and to estimate their full state [3], especially given their small size [4]. Manipulation is also complex due to DLOs high-dimensional state-space and complex dynamics [5]–[8].

Shape control of DLOs typically refers to two manipulation scenarios aiming to achieve a desired target shape. Common settings are: 1) Handling a *soft* DLO involving sequential pick-and-place actions, with surface friction holding the deformation of the DLO in place [5], [9], [10]; 2) Manipulation of *elastic* DLOs with one or more robotic arms, or with one end of the DLO fixed, allowing for better shape control, particularly in scenarios where stiffness varies or rigid/plastic behavior is involved [6], [7], [11]–[13]. Another typical differentiation within the shape control task is between model-free approaches, e.g. [13]–[16], and model-based ones, e.g. [5]–[8].

In this paper, a manipulation framework exploiting a physical prior of DLOs dynamics is proposed for shape control. A learned neural network (NN) model of the DLOs' dynamics is developed to predict the DLO behavior under manipulative actions. First, an analytical DLO model based on the mass-spring-damper formulation [17] is employed for dataset generation by systematically sampling a variety of model parameters, diverse DLO configurations, and various manipulation actions. Consequently, an NN is trained utilizing this generated dataset, i.e. training phase of Fig. 1. Notably, the NN is conditioned over several analytical model parameters, such that it can be easily adapted to match different real-world DLOs. The obtained NN model is employed during the online phase of Fig. 1 to estimate the manipulation actions to steer the DLO from its initial to a final target configuration, performing the shape control task.

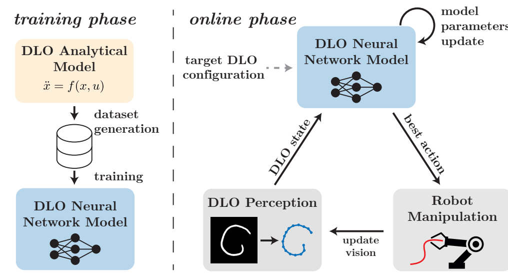


Fig. 1: Schematic overview of the manipulation framework.

The proposed framework can directly be applied for the manipulation of various DLOs on diverse surfaces, thanks to the data-driven approximation of the DLO dynamics conditioned on the model parameters. Therefore, there is no need to: 1) generate every time new task-specific data as in [12], 2) introduce complex online adaptation controllers as in [6], [7], 3) perform cumbersome and not intuitive parameters identification procedures as in [8], [18].

This paper is a compressed version of [19], we invite interested readears to refer to the full version.

II. Method

A. Analytical Model

A DLO is physically modeled using nodes with mass connected by axial springs, forming a serial chain [17] (see Fig. 2). Torsional springs at each node represent bending stiffness, while damping terms, proportional to node velocity, enhance model stability. Therefore, the dynamics of the generic node i can be written as:

$$m_i \ddot{\mathbf{p}}_i = -k_d \dot{\mathbf{p}}_i + \mathbf{f}_i^s + \mathbf{f}_i^b, \tag{1}$$

where $\mathbf{p}$ is the node coordinates, k_d a damping constant, f_i^s the force due to the axial effects and f_i^b the forces due to the bending effects, see [19] for more details.

The manipulation action executed on the DLO model is parametrized as a pick-and-place operation executed on the edge of the DLO, i.e. between two consecutive nodes. The DLO action parameters vector is defined by $a = [\alpha, \delta_x, \delta_y, \delta_\theta]$, where α denotes the index of the edge to grasp, δ_x and δ_y are the linear displacements applied to the selected edge $\{\mathbf{p}_\alpha, \mathbf{p}_{\alpha+1}\}$ and δ_θ is the rotation applied to the initial edge orientation. The effect of the action is simulated using forward Euler method applied to the discretized version of eq. (1).

Alessio Caporali, Kevin Galassi, Riccardo Zanella and Gianluca Palli are with DEI, University of Bologna, Bologna (Italy). Piotr Kicki and Krzysztof Walas are with Poznan University of Technology, Poznan (Poland).

This work is supported by the Horizon Europe project *IntelliMan* under grant number 101070136]. Corresponding author: alessio.caporali2@unibo.it

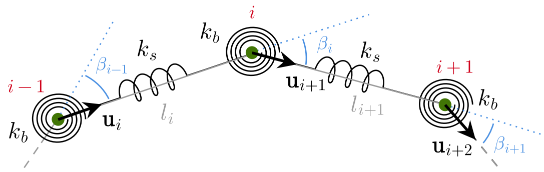


Fig. 2: DLO analytical model representation.

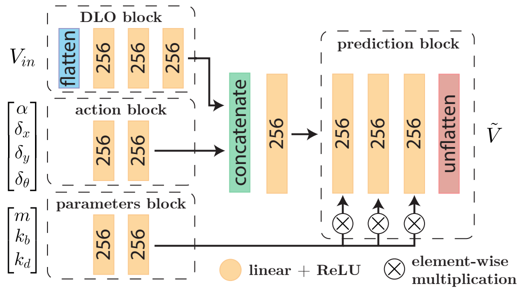


Fig. 3: Neural network architecture.

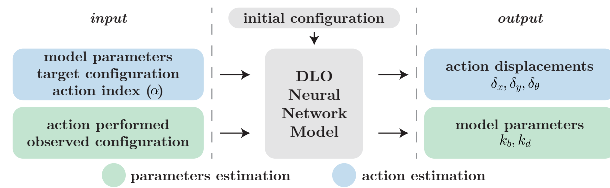


Fig. 4: Gradient-based action and DLO parameters estimation.

B. Neural Network Model

The complexity of the analytical DLO model affects its performance and makes using it in an online framework challenging. Instead, a NN can be trained to accurately replicate the DLO dynamics by exploiting a dataset of DLO movements, which can be generated offline using the analytical DLO model. Therefore, a constant and short inference time is obtained by the NN, which is more than an order of magnitude smaller than the time needed to evaluate the analytical DLO model. Additionally, the NN model is easily differentiable wrt the parameters, improving the possibility of optimizing all relevant tasks.

1) Dataset Generation: The dataset is generated by simulating the analytical DLO model subjected to a set of random actions. Each data sample consists of the DLO initial and final configurations (V_{in} and V_{out}), the performed action, and the employed model parameters. Concerning the latter, k_s is kept fixed to a high value (negligible axial deformation), instead the damping k_d, the bending k_b, the length of the DLO, and the mass of the DLO change within predefined ranges. Indeed, aiming to learn a general DLO model, both the action and model parameters are drawn from a broad range of values covering the expected real-world variability.

To generate the dataset, the physical parameters are set to random values from the physically plausible ranges, and the simulated DLO is initialized with an almost linear initial configuration. Then, a set of k actions is sampled and the behavior of DLO is simulated after applying them sequentially.

2) Data Augmentation: To improve the training efficiency and generalization capabilities of the NN model, several augmentation and normalization strategies are implemented on the data. The idea is to exploit the symmetries in the DLO data to reduce the amount of information the NN has to learn.

The normalization is performed by finding a transformation that makes V_{in} aligned to the x-axis and mean-centered, and applying it to normalize both V_{in} and V_{out}. In addition, the action parameters are scaled to be within the $[0, 1]$ range for α and $[-1, 1]$ range for the displacements. The model parameters are also normalized within the $[0, 1]$ range.

3) Neural Network Architecture: The neural network architecture is based on a set of Linear layers followed by ReLU activation functions. In detail, the network is composed of four main blocks illustrated in Fig. 3: the action block, the physical parameters block, the DLO block, and the prediction block. The input of the network is the initial configuration of the DLO V_{in}, the action parameters a, and the model parameters $p = [m, k_b, k_d]$. The output of the network, denoted as $\tilde{V}$, is the sequence of predicted changes of the 2D DLO coordinates from the initial configuration. The final predicted DLO configuration V_{pred} is expressed as $V_{\text{pred}} = \mathcal{F}(V_{\text{in}}, a, p) = \tilde{V}(V_{\text{in}}, a, p) + V_{\text{in}}$. The network is trained to minimize the mean squared error between the predicted V_{pred} and the expected V_{out} final configurations.

C. Gradient-based Estimation of Action and Parameters

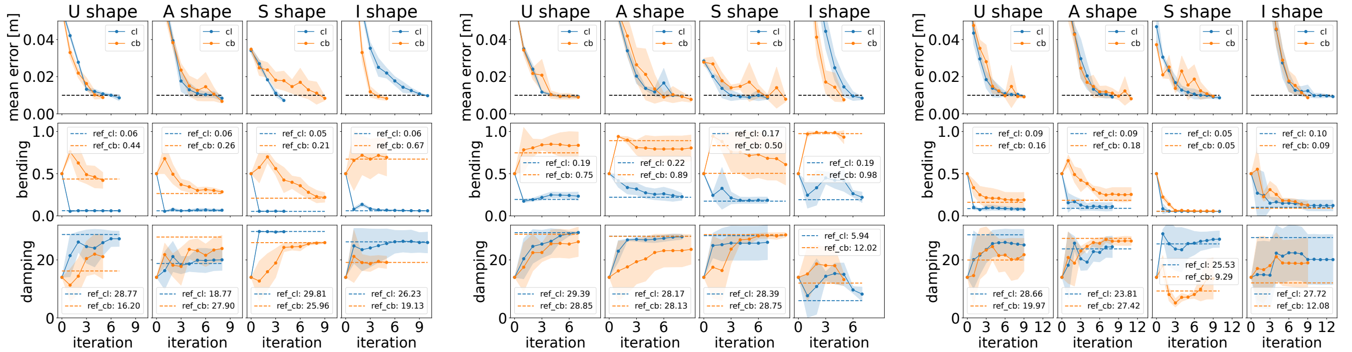
The trained NN model is used to estimate both the next manipulation action and the parameters that allow for accurate approximation of the observed DLO behavior. These two estimation procedures exploits a loss functions which is computed as the sum of L2 norms between corresponding points among two states, i.e. as $\mathcal{D}(V_1, V_2) = \sum_{i=1}^{n} \|V_{1,i} - V_{2,i}\|$. A gradient-based approach is used for the optimization of the above-mentioned loss function, see Fig. 4.

1) Action Estimation: For the best action estimation given the current DLO state V_{in} and the model parameters p, the action parameters a minimizing the difference between the NN prediction $\mathcal{F}(V_{\text{in}}, a, p)$ and the target shape V_{tgt} are sought. Therefore, the efficient batch processing capabilities of the NN model are employed, and $n-1$ optimizations are executed simultaneously, one for each edge index. Then, the best action among the ones evaluated for each edge is selected.

2) Parameters Estimation: Similarly to actions, the model parameters are estimated by searching for the ones that minimize the difference between the NN prediction V_{pred} and the observed DLO state V_{out}. Since the mass m can be measured, only k_d and k_b are estimated while m is directly provided as input to the NN model.

III. Shape Control Experiments

A robotic setup composed of a Panda Robot equipped with a parallel-jaw gripper is employed for evaluating the approach. Three ropes are used in the experiments: a *white* rope (0.45 m, 0.02 kg, 0.01 m diameter); a *black* rope (0.42 m, 0.05 kg, 0.014 m diameter); and a *red* rope (0.50 m, 0.02 kg, 0.005 m diameter). Note, the *black* rope is the stiffest one, while the *red*



(a) *white* rope (b) *black* rope (c) *red* rope

Fig. 5: Outcomes of the shape control task involving online adaptation of model parameters, conducted across various rope types and surfaces. Average results across 5 repetitions per task (standard deviations confidence region intervals). In the legends, *cl* denotes "cloth" while *cb* indicates "cardboard".

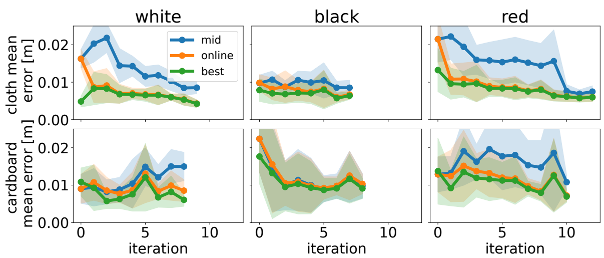


Fig. 6: Prediction errors using mid-range, online estimated, and best model parameters across ropes and surfaces.

rope exhibits a higher degree of bending elasticity compared to the *white* rope. Additionally, two planar surfaces with different physical properties are used: a *cloth* and a *cardboard* surface. The *cardboard* is smoother and more slippery than the *cloth*.

A. Shape Control Task with Online Parameters Estimation

The shape control task involves the manipulation of four distinct target shapes: *U*, *A*, *S* and *I*. The task is performed as follows. The initial DLO configuration V_{in} is a straight line, model parameters are $k_d = 14$ and $k_b = 0.5$ (mid range of values). The task is executed for each target shape V_{tgt} on each planar surface 5 times. The execution of the task is terminated once the error $\mathcal{D}(V_{\text{out}}, V_{\text{tgt}}) < 0.01$ m.

The results of the experiments are provided in Fig. 5, where, within each subplot of a specific rope, columns illustrate the task execution for specific target shapes, while rows provide an analysis of error and model parameters. In detail, the first row focuses on the mean error, with a dashed horizontal line denoting the 0.01 m threshold marking the completion of the task. The second and third rows delve into the examination of the bending parameter k_b and the damping parameter k_d respectively. Here, the dashed lines represent the estimated model parameters derived from all samples across all repetitions performed for a given shape. These values, in essence, serve as potential reference values for the specific parameters.

Analyzing the x-axis in the plots, iteration 0 represents the initial condition with a straight DLO configuration and model parameters at their initial values. An action, computed based on Sec. II-C1, is then executed by the robotic system, updating the observed DLO configuration. Model parameters are recalculated based on a single data sample (see Sec. II-C2), resulting in updated values at manipulation iteration 1. This iterative process continues until the specified termination condition is met. At manipulation iteration m, the parameter estimation is based on m data samples.

Examining the plots in Fig. 5, it is worth noting that similar bending parameters are consistently estimated for each specific rope on the *cloth* surface, regardless of the chosen target shape. The parameters estimated on the *cardboard* surface exhibit a higher degree of variability, indicating the presence of more complex dynamics due to increased slippage. The estimation of the damping term is less stable. In general, different pairs of k_d and k_b values are estimated for the same rope on different surfaces, highlighting the adaptation processes. The estimated bending parameters comparison confirms significantly different physical properties between the three ropes and that the *black* rope is the stiffest one, as initially predicted. For instance, on the *cloth* surface, the reference bending values are approximately 0.06 and 0.08 for the *white* and *red* ropes and about 0.19 for the *black* rope.

To gain a deeper insight into the impact of the online model parameters estimation, Fig. 6 presents a comparison among *mid-range*, *online* estimated, and *best* parameters. The latter refers to those estimated at the end of each task repetition, while the *mid-range* to the ones from which *online* estimation starts. These parameter setups were compared using the mean prediction error, denoted as $\mathcal{D}(V_{\text{pred}}, V_{\text{out}})$, computed after each iteration of the shape control task across all the target shapes. The plots illustrate how, within just a few iterations, the proposed method attains parameters that yield a mean error between V_{pred} and V_{out} comparable to the *best* scenario, and in most of the cases significantly better than for the *mid-range* parameters.

IV. Conclusion

The proposed manipulation framework effectively tackles shape control tasks involving various real-world DLOs and contact surfaces. It employs online parameter estimation to make the NN model predictions closely match the manipulated DLO. The efficiency of the NN model in approximating DLO dynamics is evident in both action and parameter estimation.

References

[1] A. Caporali, K. Galassi, B. L. Žagar, R. Zanella, G. Palli, and A. C. Knoll, “RT-DLO: real-time deformable linear objects instance segmentation,” *IEEE Trans. on Industrial Informatics*, 2023.

[2] A. Caporali, M. Pantano, L. Janisch, D. Regulin, G. Palli, and D. Lee, “A weakly supervised semi-automatic image labeling approach for deformable linear objects,” *IEEE Robotics and Automation Letters*, 2023.

[3] A. Caporali, K. Galassi, and G. Palli, “Deformable linear objects 3D shape estimation and tracking from multiple 2D views,” *IEEE Robotics and Automation Letters*, 2023.

[4] K. P. Cop, A. Peters, B. L. Žagar, D. Hettegger, and A. C. Knoll, “New metrics for industrial depth sensors evaluation for precise robotic applications,” in *IEEE Int. Conf. IROS*, 2021.

[5] M. Yan, Y. Zhu, N. Jin, and J. Bohg, “Self-supervised learning of state estimation for manipulating deformable linear objects,” *IEEE robotics and automation letters*, 2020.

[6] C. Wang, Y. Zhang, X. Zhang, Z. Wu, X. Zhu, S. Jin, T. Tang, and M. Tomizuka, “Offline-online learning of deformation model for cable manipulation with graph neural networks,” *IEEE Robotics and Automation Letters*, 2022.

[7] M. Yu, K. Lv, H. Zhong, S. Song, and X. Li, “Global model learning for large deformation control of elastic deformable linear objects: An efficient and adaptive approach,” *IEEE Trans. on Robotics*, 2022.

[8] N. Lv, J. Liu, and Y. Jia, “Dynamic modeling and control of deformable linear objects for single-arm and dual-arm robot manipulations,” *IEEE Trans. on Robotics*, 2022.

[9] R. Lee, M. Hamaya, T. Murooka, Y. Ijiri, and P. Corke, “Sample-efficient learning of deformable linear object manipulation in the real world through self-supervision,” *IEEE Robotics and Automation Letters*, 2021.

[10] W. Zhang, K. Schmeckpeper, P. Chaudhari, and K. Daniilidis, “Deformable linear object prediction using locally linear latent dynamics,” in *IEEE Int. Conf. ICRA*, 2021.

[11] P. Kicki, M. Bidziński, and K. Walas, “Learning quasi-static 3d models of markerless deformable linear objects for bimanual robotic manipulation,” 2023.

[12] Y. Yang, J. A. Stork, and T. Stoyanov, “Learning differentiable dynamics models for shape control of deformable linear objects,” *Robotics and Autonomous Systems*, 2022.

[13] R. Lagneau, A. Krupa, and M. Marchal, “Automatic shape control of deformable wires based on model-free visual servoing,” *IEEE Robotics and Automation Letters*, 2020.

[14] D. Navarro-Alarcon, H. M. Yip, Z. Wang, Y.-H. Liu, F. Zhong, T. Zhang, and P. Li, “Automatic 3-d manipulation of soft objects by robotic arms with an adaptive deformation model,” *IEEE Trans. on Robotics*, 2016.

[15] J. Zhu, B. Navarro, P. Fraisse, A. Crosnier, and A. Cherubini, “Dual-arm robotic manipulation of flexible cables,” in *IEEE Int. Conf. IROS*, 2018.

[16] O. Aghajanzadeh, M. Aranda, J. A. Corrales Ramon, C. Cariou, R. Lenain, and Y. Mezouar, “Adaptive deformation control for elastic linear objects,” *Frontiers in Robotics and AI*, 2022.

[17] N. Lv, J. Liu, X. Ding, J. Liu, H. Lin, and J. Ma, “Physically based real-time interactive assembly simulation of cable harness,” *Journal of Manufacturing Systems*, 2017.

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

[19] 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.