REMODEL - Robotic tEchnologies for the Manipulation of cOmplex DeformablE Linear objects

Deliverable 5.5 – Interactive Perception

Version 2023-08-31

Project acronym: REMODEL
Project title: Robotic tEchnologies for the Manipulation of cOmplex DeformablE Linear objects
Grant Agreement No.: 870133
Objects Topic: DT-FOF-12-2019
Call Identifier: H2020-NMBP-TR-IND-2018-2020
Type of Action: RIA
Project duration: 48 months
Project start date: 01/11/2019
Work Package: WP5 – Cable Manipulation Planning, Execution and Interactive Perception
Lead Beneficiary: PUT
Authors: PUT, UNIBO
Dissemination level: Public
Contractual delivery date: 31/08/2023
Actual delivery date: 31/08/2023
Project website address: https://remodel-project.eu

List of Abbreviations

Acronym	**Meaning**	**Explanation**
DLO(s)	Deformable Linear Object(s)	Highly deformable objects with an infinite amount of degrees of freedom, such as ropes, cables, wires, tubes, etc.
NN	Neural Network	A method in artificial intelligence that processes a input datum based on a set of weights that were learned in advanced over a set of examples.
CNN	Convolutional Neural Network	A type of Neural Network used for processing 2D data like images.
CK	Chroma Key	A technique used to isolate a given color range from an image or video.
IoU	Intersection over Union	Metric commonly used in image detection and segmentation to assess the accuracy of a solution or method.
ReLU	Rectified Linear Unit	A popular activation function used in neural networks, which implements the function f(x) = max(x, 0)

1 Introduction

The manufacturing processes involving Deformable Linear Objects (DLOs) like cables, hoses, and wiring harnesses are highly complex, presenting various challenges from two main perspectives: perception and manipulation. From a perception standpoint, dealing with DLOs is a tough task. Their ambiguity can make it difficult to distinguish different parts of them or to distinguish DLOs from other objects, especially given their relatively small size. Regarding manipulation, DLOs pose a formidable obstacle due to their unpredictable configuration behavior, necessitating a deep understanding of their physical characteristics to predict and control their shape. DLO manufacturing processes typically occur in a human-only setting, since, considering the challenges mentioned, nowadays it is more practical to rely directly on human expertise rather than on robotic systems.

In the deliverable D4.3 Real-Time Cable Detection and Tracking, we presented a couple of solutions to robustly detect and track DLOs in a real-time manner. However, D4.3 is addressing only the perception challenges.

Therefore, the purpose of this deliverable is to present and demonstrate solutions where the perception of the DLOs is exploited in a complex robotic system to accomplish difficult tasks. The deliverable presents two main tasks: efficient labeling of real-world DLOs images; DLOs manipulation planning.

The first task addresses the problem emerging with recent deep-learning methods requesting huge datasets for their optimization. Images of DLOs require an enormous human effort for their labeling due to the intrinsic peculiarities of DLOs. Synthetic approaches can be employed to avoid this tedious task, but they introduce other difficulties like synthetic to real gap. Thus, being able to label real-world images of DLOs efficiently is a big benefit. The efficiency is accomplished by an interaction between a human operator, annotating just a few points along a DLO, and a perception system, able to extrapolate fully dense and precise labels starting from the operator input. This approach is presented in detail in Sec. 2.

DLOs manipulation planning is a complex task, where simple approaches commonly used for rigid objects do not apply effectively because of the mentioned challenges concerning DLOs manipulation behavior. In this deliverable, a framework combining the perception system and the manipulation system is exploited to improve the manipulation capabilities. The goal of the manipulation planning framework is estimating manipulation actions able to steer the DLO from its initial to a final target configuration. This framework is detailed in Sec. 3.

2 Weakly Supervised Annotation of DLOs

Existing data-driven approaches, in real-world applications, are still massively affected by the quality and size of the datasets. Although some works have already addressed the problem of generation of data (for instance employing the CK method[1]), there are no methods that have dealt efficiently with real images. Indeed, the problem of employing real images in data-driven approaches concerns the annotation phase. A tedious and long labeling process is required, where the utilization of human work is not efficient. This section describes an approach developed to tackle the efficient labeling of real-world data by exploiting an interactive approach. A more detailed discussion is available in the associated publication[2].

To enable a more efficient labeling of DLOs in a real environment, an interactive approach making use of an infrared tracker system is exploited. The idea is to have a human operator annotating only keypoints along the DLOs in the workspace scene. Then, a robotic arm with a camera mounted on the gripper is used to collect a huge set of images of the annotation scene. Finally, by means of the knowledge of the camera location, each image is properly annotated with the collected initial keypoints. The generated image labels are corrected to account for possible calibration and user input errors exploiting a CNN specifically developed. Finally, the keypoints labels are converted to dense pixel-wise masks. For brevity, the proposed method is denoted as DLO-WSL standing for DLO Weakly Supervised Labeling. Figure 1 schematizes the proposed approach.

The labeling of the DLOs instances via an infrared tracker approach is detailed in Sec. 2.1. The recording of the set of images is outlined in Sec. 2.2. The projection of the labeled keypoints in each image sample is described in Sec. 2.3 and the correction and mask generation procedures are presented in Sec. 2.4. Finally, in Sec. 2.5 the experimental validation of the proposed approach is reported.

2.1 Infrared Tracker Labeling

A methodology involving a sensor tracked in space is selected to label instances of DLOs. The Tracepen device is selected. It works on the basis of reflective photodiode sensors which, by receiving and mirroring an infrared signal, enable the calculation of the sensor's pose from the emitting station. Due to this working

[1] R. Zanella, A. Caporali, K. Tadaka, D. De Gregorio and G. Palli, "Auto-generated Wires Dataset for Semantic Segmentation with Domain-Independence," International Conference on Computer, Control and Robotics (ICCCR), 2021

[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," in *IEEE Robotics and Automation Letters* 2023

principle, the coordinates of the tracker are expressed in reference to the emitting station. However, to obtain DLOs instance labels for the images, such positions had to be transformed in camera pixel coordinates. To solve this issue, homogeneous transformations between the emitting station and the camera position had to be considered. Hence, the transformation between the emitting station and the robot is calculated to obtain tracker points in the robot coordinate frame.

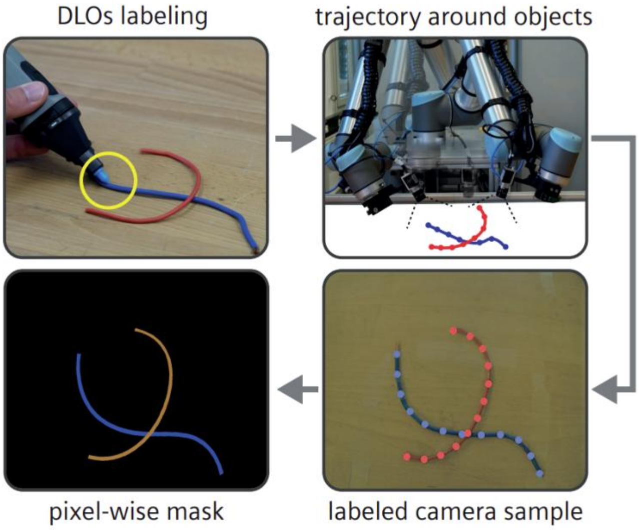


Figure 1: The DLO-WSL approach.

2.2 Recording of Images with a Robot

To create a dataset via DLO-WSL, images along with the position of the camera in the world coordinate system are needed. Therefore, a 2D RGB camera is mounted on the flange of a robotic arm in an eye-in-hand configuration for a two-fold benefit. First, the recording of several images is achieved in a matter of seconds. Second, the position of the camera is known as long mechanically connected to the robot. A robot trajectory with an ellipsoidal shape is integrated into the robot control to ensure that the images taken by an inward-facing camera had always the object at the center of the trajectory. This trajectory is calculated as:

$$\begin{cases} x = a \sin\theta \sin\phi + x_0 \\ y = -b \sin\theta \sin\theta + y_0 \\ z = c \cos\theta + z_0 \end{cases}$$

where x, y, z are the trajectory points, a, b, c are the ellipsoid parameters, θ is the zenith angle, ϕ is the azimuth angle, and x0, y0, z0 are the coordinates of the initial position.

2.3 Projection Keypoints

The collected DLOs labels are expressed in the 3D cartesian frame of the robot. They are projected in the 2D images to generate training data labels. This is achieved through the following equation:

$$\begin{bmatrix} u' \\ v' \\ w' \end{bmatrix} = K\,{}^{r}T_c \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

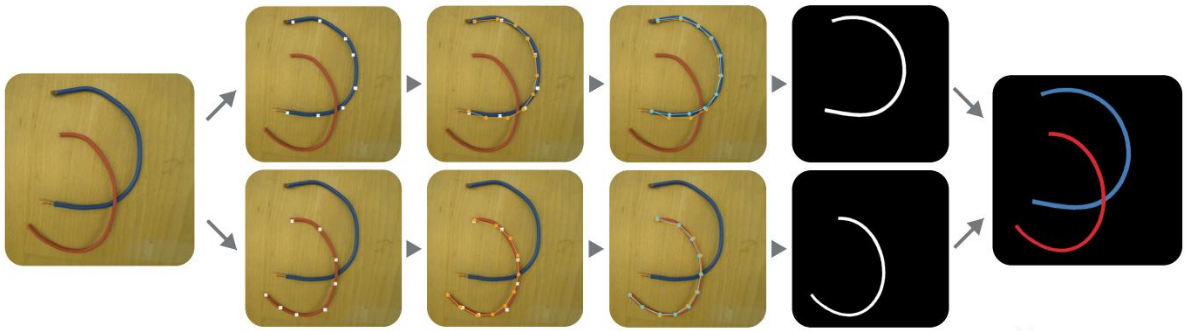
where u, v are pixel coordinates, w′ is the scaling factor, K is the intrinsic camera matrix obtained via camera calibration, ${}^{r}T_{c}$ is the transformation from robot to camera, [x, y, z] is the 3D point which needs to be projected, and u = u ′/w′ and v = v′/w′.

With this approach, images of real-world scenarios with different camera positions can be labeled using a single initial labeling input. This procedure is repeated for each DLO instance to label.

2.4 Correction Procedure

Unlike synthetic labeling, which is inherently error-free, during the labeling of real-world DLOs performed by a human operator, some level of error is expected. In particular, the major sources of errors are due to inaccuracies in 1) the calibration of the annotation tool and/or eye-in-hand camera; 2) the labeling performed by the human operator. The presence of errors is more evident and severe especially on very thin DLOs.

To overcome these problems, a fine-tuning step is applied after the human input to each labeled DLO instance, its main stages are shown in Figure 2.



(a) input image (b) VR input (c) spline (d) correction (e) instance mask (f) output mask

Figure 2: Correction procedure developed within DLO-WSL.

First, the labeling points of one DLO instance are smoothed employing an approximating spline curve in the 2D pixel space. Then, an approach based on a

CNN is applied for computing a correction offset for each labeled point. The CNN-based network performs the computation illustrated in Figure 3.

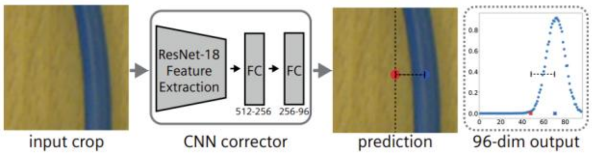


Figure 3: CNN-based corrector employed in DLO-WSL.

Given the source image, the vertically oriented crop extracted around the considered labeled point is extracted. With vertically oriented, we denote the condition with the DLO having an almost vertical shape. Thus, the CNN-based network approximates the location of the DLO in the image along the horizontal axis. In other words, the probability of each image column of corresponding to the DLO centerline is computed and the peak of the obtained curve selected as DLO center.

Having corrected the points, the knowledge of the DLO thickness is exploited to construct a polygon that precisely follows the contour of the targeted DLO in each image plane. Thereafter, from the polygon, an instance mask can be easily drawn, as shown on the right-hand side of Figure 2.

2.5 Experiments

To evaluate the capabilities of the proposed DLO-WSL, a user test with a balanced randomized order of three subsequent interactions was envisioned. Since no labeling method exists for big datasets which requires labeling for only one image with uneven backgrounds, the comparison is done against the CK technique for its adequacy to generate multi-image datasets with single human intervention in even backgrounds, and RITM[3] due to its good performance in state-of-the-art single image weakly supervised labeling.

For these comparisons, the users were requested to label 10 images with the three different methods. At the end of each interaction, usability and workload were measured through the System Usability Scale (SUS)[4] and the NASA-TLX[5] Additionally, the number of clicks (NoC) to complete the labeling task was also recorded. A total of 13 users, not experienced with labeling techniques, age mean

[3] K. Sofiiuk, I. A. Petrov, and A. Konushin, “Reviving iterative training with mask guidance for interactive segmentation,” in 2022 IEEE International Conference on Image Processing (ICIP).

[4] J. Brooke, SUS - A quick and dirty usability scale. CRC Press, 1996.

[5] S. G. Hart and L. E. Staveland, Development of NASA-TLX (Task Load Index): Results of Empirical and Theoretical Research. Elsevier, 1988.

(M) = 32.70 years, standard deviation (SD) = 9.23, participated in the study. All of them performed the test correctly and no data was discarded.

Concerning the usability, the score for CK is M=60.38%, SD=21.00, for DLO-WSL is M=69.61%, SD=16.26, and for RITM is M=82.30%, SD=9.54. From a statistical standpoint, it is possible to conclude that the usability of DLO-WSL is good and comparable to CK and RITM. Regarding perceived effort, the score for CK is M=30.51%, SD=15.08, DLO-WSL is M=29.74%, SD=12.96, and for RITM is M=22.31%, SD=13.12. Thus, it is possible to conclude that the workload perceived in using DLO-WSL is comparable to the other methods. In Figure 4 a boxplot illustrating both the usability and effort results is provided.

Finally, to examine the performances of labeling, the average IoU and IoU over the average Number of Clicks (NoC) for the dataset of 10 images are used. The results are shown in Figure 5. More precisely, IoU scores were M=91.68% SD=6.56 for CK, M=91.32% SD=1.52 for RITM, M=81.05% SD=6.22 for DLO-WSL and M=36.88% SD=12.45 for spline. IoU/NoC scores were M=15.31%/clicks SD=1.09 for CK, M=6.54% SD=2.78 for DLO-WSL, M=3.16%/clicks SD=2.05 for spline and M=0.21%/clicks SD=0.13 for RITM. Thus, DLO-WSL obtains a good average IoU while minimizing the number of clicks for uneven backgrounds.

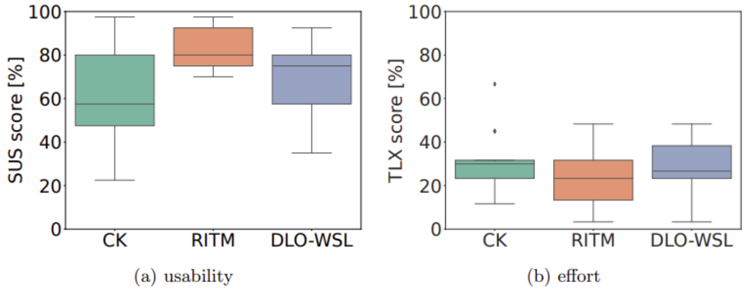
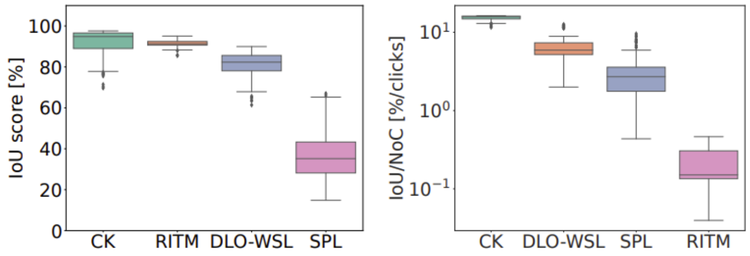


Figure 4: In (a) usability measure with the SUS scale, the higher the better. In (b) workload measure with the NASA TLX, the lower the better. The comparison is established between chroma-key (CK), RITM and DLO-WSL.



(a) segmentation

(b) segmentation and number of clicks

Figure 5: Evaluation of the performances in terms of semantic segmentation and number of clicks for the labeling approaches. The segmentation reports the outcomes with chroma-key (CK), RITM, DLO-WSL and raw data from the user (without the CNN correction) fitted with a spline (SPL).

3 Manipulation Planning via Interactive Perception

In this section, perception and manipulation are combined into a unified framework where perception, done during manipulation, helps to improve the manipulation capabilities of a robotic system. In particular, the DLO behavior is approximated via a *model* predicting the DLO shape after a manipulation action. The model prediction is conditioned over a set of physical parameters of the DLO (e.g., mass, bending elasticity, etc.).

Thus, the perception system is responsible for providing accurate detections of the DLO shape such that: 1) the physical parameters can be estimated via an optimization process performed on the model; 2) the model can be initialized with accurate initial conditions of the DLO shape, and an effective manipulation action can be predicted by the model.

The introduced *model* can be either an analytical model obtained as a set of differential equations, or a specialized NN designed to approximate the analytical model. Indeed, NNs are known to be universal function approximators given a proper set of training data to be optimized against. Nevertheless, the presence of the model is crucial as it enables exploring possible manipulation actions and selecting the best one to achieve a desired final configuration of the DLO.

Therefore, an interactive framework between perception and manipulation originates. The overall method comprises four main elements, which are:

- Perception system;
- DLO analytical model;
- DLO neural network model;
- Manipulation system.

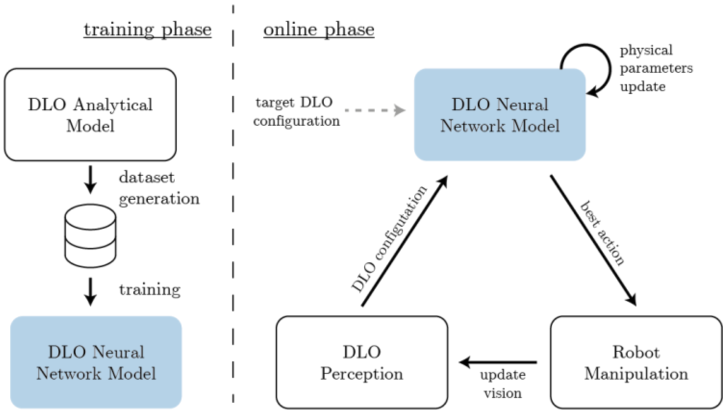


Figure 6: Interactive perception framework.

As depicted in Figure 6, the framework exploits two different phases, namely a training phase and an online phase. In the first the DLO analytical model is used for the generation of a training dataset which is employed to optimize the weights of the NN model.

In the online phase, only the DLO NN model is utilized. This choice comes from computation-related reasons: the analytical model requires careful choices of integration time and simulation steps that heavily affect the model performances. Instead, the NN model conditioned over a wide range of meaningful set of physical parameters and initial/target DLO shapes, can accurately approximate the analytical model while gaining a significant boost in processing time. This improved efficiency allows the utilization of the NN model in the online phase.

In the online phase, the DLO NN model is responsible for accomplishing two main tasks:

- Physical parameters estimation;
- Best manipulation action estimation.

The first consists of estimating the physical parameters of the DLO affecting the final prediction. The latter is related to the estimation of the best possible action to be performed via a robotic arm (the manipulation system) toward a target DLO shape configuration.

In the rest of this document, the perception system, DLO analytical model and DLO NN model are analyzed in Secs. 3.1, 3.2 and 3.3 respectively. The estimation tasks are instead presented in Sec. 3.4. Finally, Sec. 3.5 discusses the obtained experimental results.

3.1 DLO Perception

The perception of the DLO is accomplished exploiting the results achieved in the Deliverable 4.3 and published as journal paper[6] with an algorithm named RT-DLO (see Figure 7).

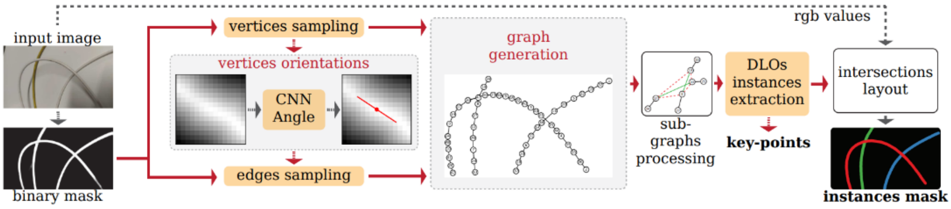


Figure 7: The RT-DLO algorithm employed for the DLOs perception

A Photoneo Motioncam3D sensor is statically mounted on a robotic cell providing point-cloud data. The plane characterizing the workspace is easily segmented from the cloud obtaining all the points describing the DLO. The points are projected on the image plane by utilizing the camera's intrinsic parameters d a binary mask of the DLO is obtained. In the mask, the pixels belonging to the DLO have a value of 1, while all the others have a value of 0. The binary mask is forwarded to RT-DLO that performs the segmentation and modelling of each DLO instance present in the scene. Each DLO is modelled as a sequence of nodes and vertices, effectively obtaining as output a line graph representation of the DLO as shown in Figure 8.

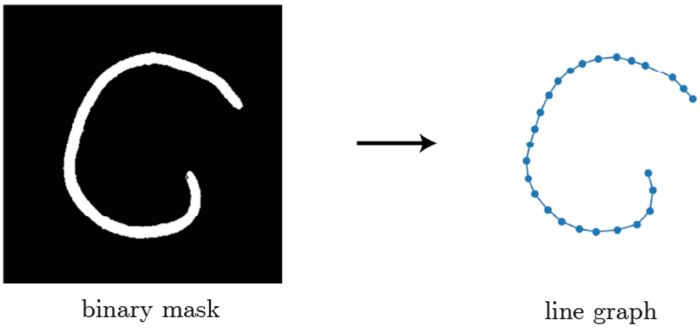


Figure 8: Detected DLO described as a line graph obtained from a binary mask.

The line graph characterization of the DLOs is thus exploited by the mathematical DLO analytical model and neural networks model, as detailed in Secs. 3.2 and 3.3.

3.2 DLO Model

A DLO is physically modeled via masses and springs. Considering the output of the perception pipeline, a point mass is placed at every node location while the edges between the nodes are replaced by springs. These springs are used to model the axial effects of DLOs. In addition to point masses and axial springs, also the bending

[6] A. Caporali, K. Galassi, B. L. Žagar, R. Zanella, G. Palli and A. C. Knoll, "RT-DLO: Real-Time Deformable Linear Objects Instance Segmentation," in *IEEE Transactions on Industrial Informatics, 2023*.

effect is modeled by placing a torsional spring at each node. The mathematical model is thus a set of point masses connected by axial and torsional springs. To better simulate the DLO behavior, the model is augmented with a damping term. The damping term is modeled as a viscous friction force proportional to the velocity of the point mass.

An illustration of the masses and springs insertion from the line graph is shown in Figure 9. For a sample node *i*, its mass is denoted by m_i while the axial and torsional springs have constants k_s and k_b respectively.

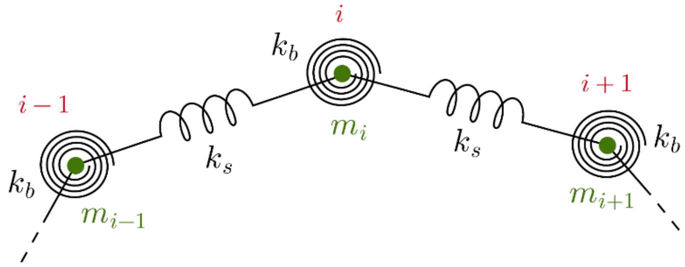


Figure 9: DLO model with masses, axial spring and torsional springs.

The motion law of the mass points *i* can be determined by Newton's second law as:

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

where p_i 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. The axial effects are computed as:

$$\mathbf{f}_i^s = -k_s(l_i - l_i^0)\mathbf{u}_i + k_s(l_{i+1} - l_{i+1}^0)\mathbf{u}_{i+1}$$

where l_i and l_i^0 are the current and initial lengths of link *i* respectively. With u_i the unit vector of node *i* is denoted. The bending effects are computed as:

$$\begin{aligned}\mathbf{f}_i^b =& k_b \frac{\beta_{i-1}}{l_i \sin \beta_{i-1}} \mathbf{u}_i \times (\mathbf{u}_{i-1} \times \mathbf{u}_i) \\ & - k_b \frac{\beta_i}{l_i \sin \beta_i} \mathbf{u}_i \times (\mathbf{u}_i \times \mathbf{u}_{i+1}) \\ & - k_b \frac{\beta_i}{l_{i+1} \sin \beta_i} \mathbf{u}_{i+1} \times (\mathbf{u}_i \times \mathbf{u}_{i+1}) \\ & + k_b \frac{\beta_{i+1}}{l_{i+1} \sin \beta_{i+1}} \mathbf{u}_{i+1} \times (\mathbf{u}_{i+1} \times \mathbf{u}_{i+2})\end{aligned}$$

where:

$$\beta_i = \arctan \frac{||\mathbf{u}_{i+1} \times \mathbf{u}_i||}{\langle \mathbf{u}_{i+1}, \mathbf{u}_i \rangle}$$

is the angle between link *i* and *i+1.*

In Figure 10 the bending computation schematic is illustrated via a geometry description of a DLO portion.

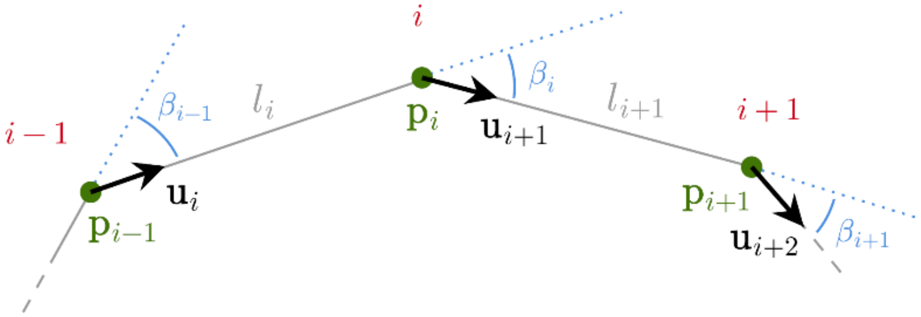


Figure 10: Geometry of a section of a DLO.

The goal of the DLO model is to predict the future position of the DLO nodes given a manipulation action. It is assumed that a pick and place action takes place somewhere on the DLO. The action is assumed to be confined on the plane and it is described by the following parameters: edge index, edge x-coordinate displacement, edge y-coordinate displacement and edge z-axis rotation angle differenece. Indeed, since the DLO is modeled as a line graph, the action is described by the edge index where the action takes place and the displacement subject by the edge from its initial configuration. The choice of performing the action at the edge level and not on the individual node is motivated by the fact that the action is performed by a gripper that, during the grasp, interacts with the DLO on a given section that is better described by an edge rather than by a point mass node.

3.3 Neural Network Model

A neural network is employed to approximate the DLO model and gain a significant boost in terms of computational efficiency. Indeed, the DLO model complexity affects performances making its utilization in an online framework challenging. Instead, a neural network can exploit a dataset generated offline by the DLO model to accurately approximate it while effectively improving the time performance by order of magnitude.

Thus, utilizing the DLO model, a dataset of DLO motions is generated and collected as detailed in Section 3.3.1. Thereafter, a neural network, having the structure defined in Section 3.3.2, is trained as detailed in Section 3.3.3.

3.3.1 Dataset Generation

The dataset to be used for the training of the neural network is generated by simulating the DLO model performing a set of random actions. The dataset consists of the following quantities that are obtained by the DLO model and saved: DLO initial configuration, DLO final configuration, action commanded, and physical parameters.

The DLO initial and final configuration is just the set of 2D locations on the cartesian space of the nodes constituting the DLO. Thus, assuming that the DLO is modelled by n nodes, each shape has dimensions $\mathbb{R}^{n\times2}$.

The action is described by the same parameters of Sec. 3.2, namely the edge index (where the action takes place) and the motion offset (along x and y axis, rotation on the z axis).

Finally, the physical parameters specify the type of behavior achieved by the DLO model. These parameters are: the damping term K_d, the bending term K_b, the length of the DLO, and the mass of the DLO. Both the length and the mass are assumed to be known quantities since they can be usually measured. The other two terms are instead more difficult to measure and should be estimated in alternative ways.

Accounting for the selection of the physical parameters, the set of values assigned to the 4 parameters are sampled from a uniform distribution. The range of the distribution is selected to cover the entire range of possible meaningful values. Thus, the damping term (K_d) is selected from the range (1, 30) while the bending term (K_b) from the range (0.05, 1.5). The length is specified between 15 and 50 centimeters. The mass is confined between 10 and 100 grams.

Having specified the parameters and sampled a random initial configuration, it is now required to sample a random action to be commanded to the DLO. Thus, a random edge index is selected specifying the location of actuation. Thereafter, given a minimum and maximum range of displacement and an action direction, the linear displacement variable (x, y) are computed. Finally, a random value for the angular action is sampled from a uniform distribution as well. Thus, the combined action is commanded to the mathematical model. The displacement range is set to be between 5 and 10 centimeters while the action direction angle range is set to be within 90 degrees.

The mathematical model is simulated for a fixed number of simulation steps and the final configuration reached by the DLO is saved. An example of a data sample contained in the generated dataset is illustrated in Figure 11.

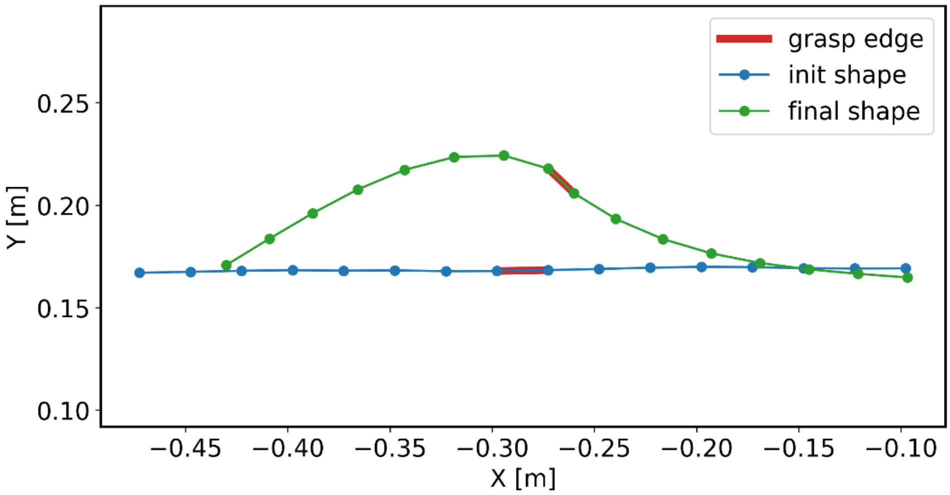


Figure 11: Simulation of a DLO manipulation performed at edge index 7 with displacement (0.025, 0.05) m and rotation 30 degrees.

3.3.2 Network architecture

The neural network architecture is based on a set of Linear layers followed by ReLU activation functions. The input of the network is the initial configuration of the DLO, the action parameters, and the DLO's physical parameters. The output of the network is a sequence of predicted changes of the 2D DLO coordinates between the initial and final configuration of the DLO.

The network is composed of four main blocks: 1) the action block, 2) the physical parameters block, 3) the DLO block, and 4) the prediction block.

In Figure 12 the architecture of the network is shown.

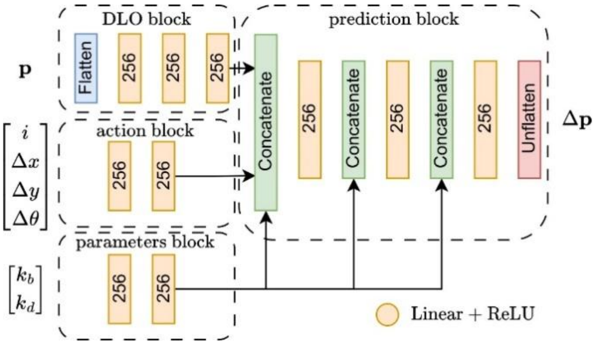


Figure 12: Neural network structure

3.3.3 Network training and error metrics

The network is trained to minimize the mean squared error between the predicted and the ground truth change between the initial and final configuration.

In Figure 13 a plot describing the training progress over time is shown. The neural network is able to capture the DLO dynamics simulated by the model reaching a final error on the validation set of less than 2 millimeters.

As error metric, the point-to-point distance between the target DLO configuration and the predicted one is employed.

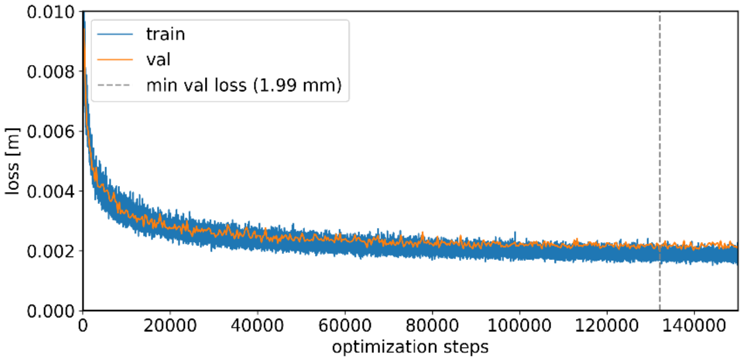


Figure 13: Train and validation loss of the neural network. The dashed grey line indicates the location of the minimum validation loss.

3.4 Exploiting the Neural Network for Prediction

In order to properly predict the future configuration of the DLO, estimates close to the real physical parameters of the DLO are needed. Indeed, the physical parameters condition directly the DLO behavior and thus the predicted configuration. The trained

Neural Network DLO Model can thus be employed to accomplish different tasks based on which input/outputs setup is used, as described by Figure 14.

In particular, the two important processes exploited are: the estimation of the real physical parameters of the DLO that is presented in more details in Sec. 3.4.1; the estimation of the action to be performed in order to achieve the desired target configuration, described in Sec. 3.4.2.

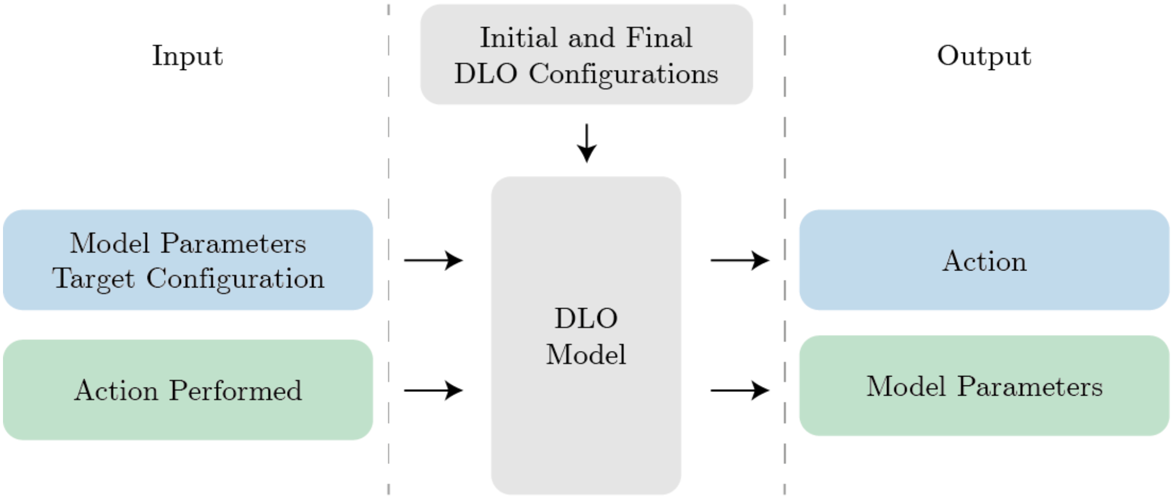

Input
Initial and Final DLO Configurations
Output
Model Parameters Target Configuration
Action Performed
DLO Model
Action
Model Parameters

Figure 14: Action and parameters estimation from the point of view of the DLO model input/output.

3.4.1 Physical Parameters Estimation

The proposed approach to modeling the DLO behavior using a neural network enables us to easily estimate the parameters of the DLO. In particular, there are two simple methods to do so: 1) using forward pass, and 2) using backward pass through the network. The first method consists of repeatedly asking the NN-based DLO model about the result of applying a given action to the DLO with different sets of parameters and comparing these results with the actual DLO displacement. Thanks to the short time of the neural network inference, even such a trivial approach can be effective if the set of parameters is sufficiently small (which is true in the considered case). A more sophisticated approach to the problem of estimation of the physical parameters is to exploit the innate differentiability of the considered neural network, and instead of taking the gradient with respect to the weights of the neural network as it is done during the training, one can differentiate the prediction error with respect to physical parameters, and update them based on this gradient. By updating these parameters repetitively, it is possible to reach a stable minimum of the loss, and thus find the set of the parameters that explains the observations the best.

The approaches described above, thanks to the native parallelism of the neural network inference and back-propagation, enable fast parameter estimation not only for a single observation but also for a batch of experiments. By applying estimation to a batch of action-observation pairs one can estimate parameter values much more reliably without significant overhead.

3.4.2 Best Action Estimation

In the previous section, we introduced a way of utilizing the trained neural network to estimate the parameters of the DLO. Having the set of parameters already estimated, one can use a similar approach to estimate the best action. In this case, the idea is to find an action that minimizes the error between the predicted state of the DLO and the desired one.

In the considered approach, the action consists not only of the continuous part, which may be easily adapted using the gradient of the aforementioned error but also of the integer value of the node to which the action is applied. To handle this type of variable, we once again relied on the ability of the neural networks to be easily parallelized and simply performed n-1 independent optimizations, where n is a number of the vertices of the DLO representation, and we choose the action that resulted in the smallest difference between the prediction and the desired DLO state.

3.5 Experiments

The experiments have been carried out to assess the capabilities of the proposed framework. Two ropes with similar material and different lengths (30 and 44 cm) are used. Each experiment is initiated with the rope placed in a straight configuration. The goal is to reach a target configuration in a given amount of manipulation steps (or trials). Since the NN model is conditioned over the DLO parameters (K_b, K_d and mass), prior to the manipulation sequence a parameters estimation phase is exploited for computing the damping and bending coefficients. The mass is instead provided directly by measuring the DLO.

The parameters and action estimations are computed employing the methodology discussed in Sec. 3.4.

3.5.1 Parameters Estimation

The parameters are estimated by collecting 5 random manipulation actions on each rope. Thus, to the NN model, the following quantities are forwarded: initial DLO configuration, final DLO configuration, action performed. Additionally, the mass of the DLO is measured and provided directly. The NN optimizes the damping and bending coefficients to reduce the overall loss between the predicted NN shape of the DLO and the final shape reached by the real system.

In Figure 15 and Figure 16 are depicted the plots of the coefficients and loss curves for the two ropes. From the plots it is clear that in both cases the coefficients converge, and the overall loss is reduced. In Table 1, the estimated values are provided.

Table 1: Estimated parameters for the two ropes.

	K_d	K_b
Rope 30 cm	8.74	0.11
Rope 44 cm	8.95	0.10

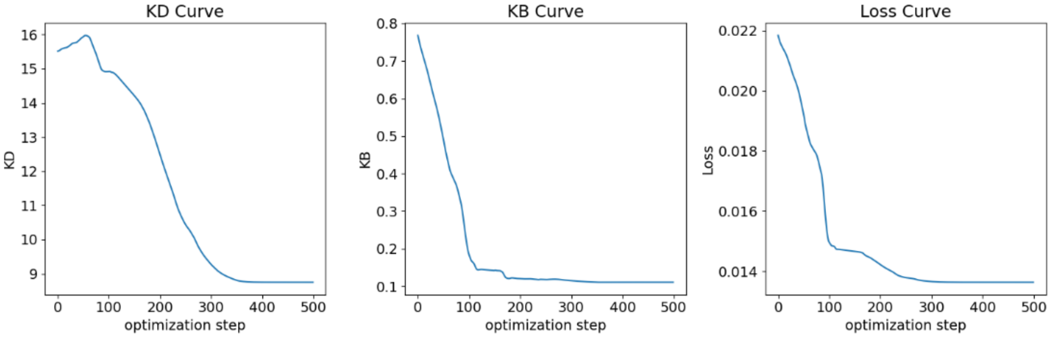


Figure 15: Parameters estimation for 30 cm rope.

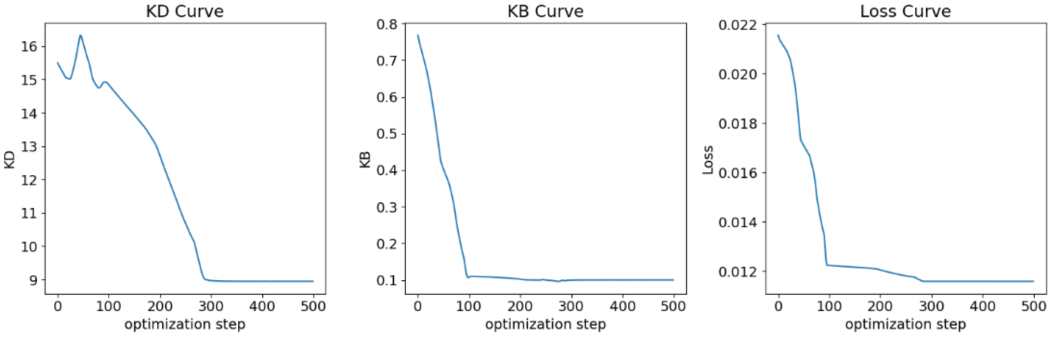


Figure 16: Parameters estimation for 44 cm rope.

3.5.2 Action Estimation

Given the estimated parameters, the experiments related to the manipulation sequence aiming at reaching the final target configuration are shown here.

To better analyze the behavior of the NN in predicting the action to be performed, different values of parameters are employed. In particular, a comparison between the best (the ones estimated) and mid (the ones in the middle of the considered parameter ranges) is established. The mid parameters are set to: K_d equals to 15, K_b equals to 0.75.

As target shapes, the “I”, “L”, “S” and “U” configurations are considered. From Figure 17 to Figure 20 are shown the results for the 44 cm case. Instead, from Figure 21 to Figure 24 are depicted the results for the 30 cm case.

The NN with the best parameters shows both a better progression toward the final goal and a faster convergence toward smaller errors. This is noticeable more on the 44 cm rope rather than the shorter 30 cm one, as expected due to the longer shape and thus bigger differences.

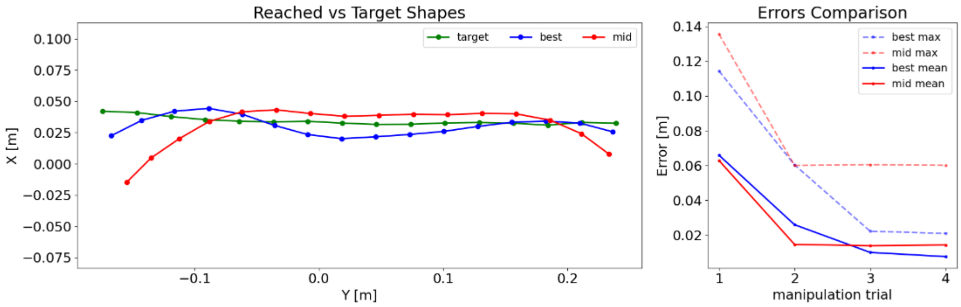


Figure 17: Reaching "I" shape with a 44 cm long rope**.**

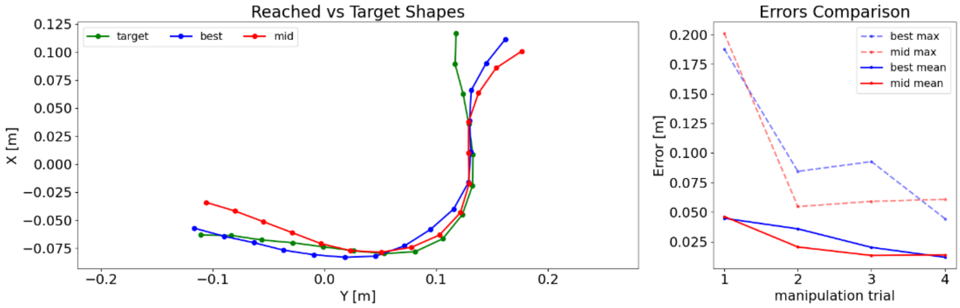


Figure 18: Reaching "L" shape with a 44 cm long rope**.**

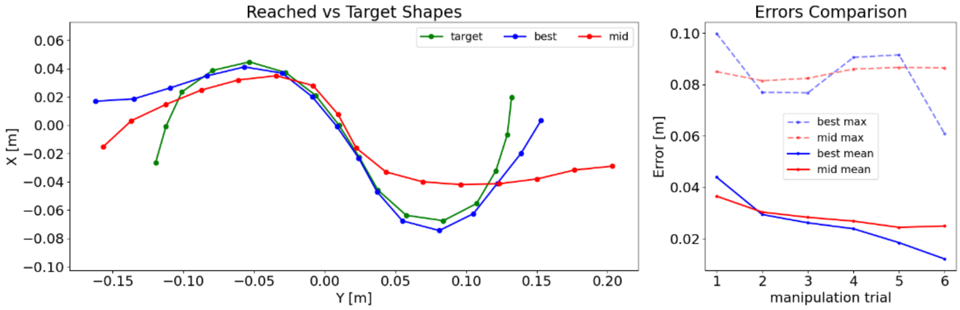


Figure 19: Reaching "S" shape with a 44 cm long rope**.**

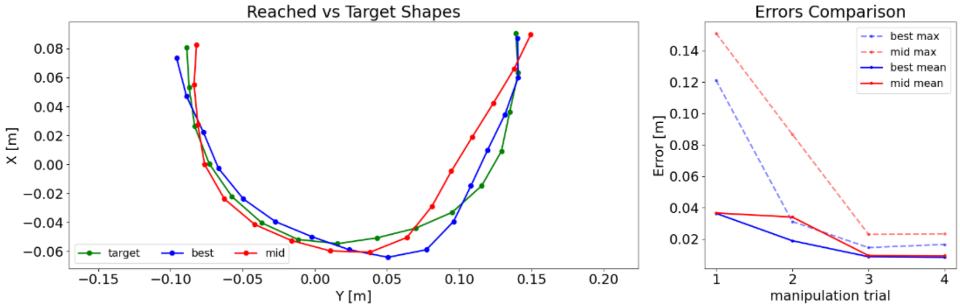


Figure 20: Reaching "U" shape with a 44 cm long rope**.**

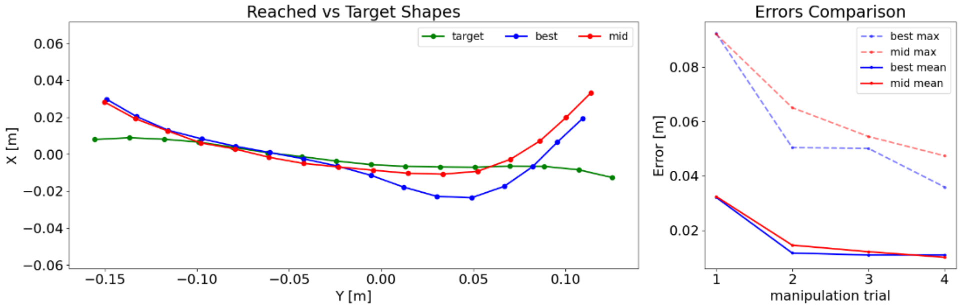


Figure 21: Reaching "I" shape with a 30 cm long rope**.**

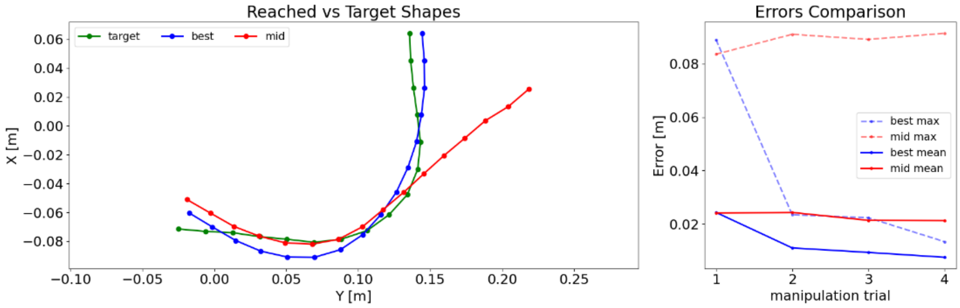


Figure 22: Reaching "L" shape with a 30 cm long rope**.**

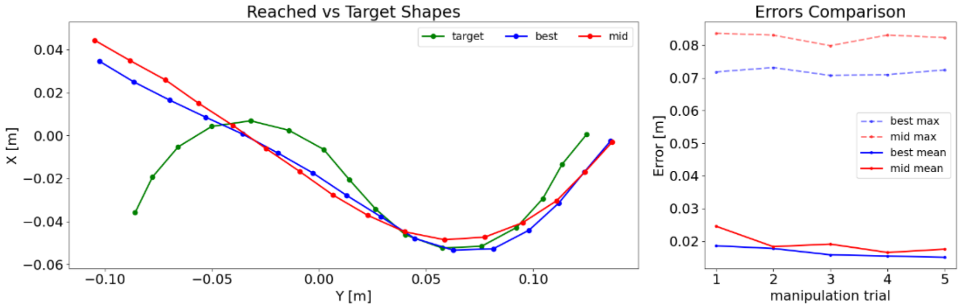


Figure 23: Reaching "S" shape with a 30 cm long rope**.**

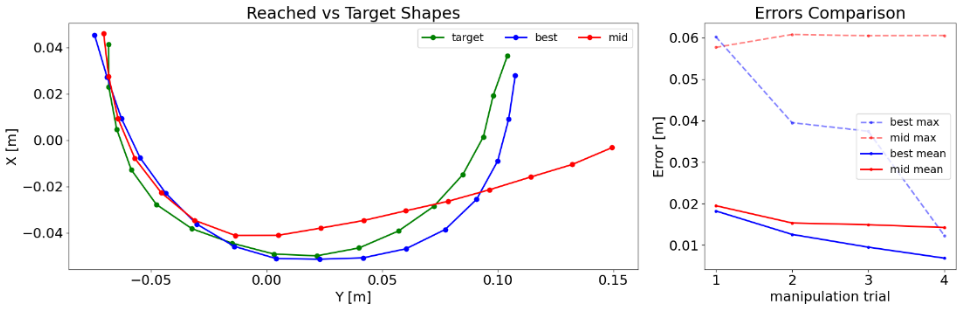


Figure 24: Reaching "U" shape with a 30 cm long rope**.**

4 Summary

In this deliverable several solutions were presented concerning the exploitation of the perception of DLOs in an interactive fashion for accomplishing different tasks like efficient labelling of real-world images or best manipulation action planning. The efficient labelling technique proves to be effective compared to existing approaches. The best manipulation action planning involves the contribution of several systems along the perception one, demonstrating the challenges involved with DLOs manipulation. Nevertheless, the experiments demonstrated the capabilities of the proposed framework in achieving different target shapes with DLO of different lengths.