REMODEL - Robotic tEchnologies

for the Manipulation of cOmplex

DeformablE Linear objects

Deliverable 5.3 – Bimanual cable manipulation strategy

Version 2019-10-14

Project acronym: REMODEL
Project title: Robotic tEchnologies for the Manipulation of cOmplex DeformablE Linear objects
Grant Agreement No.: 870133
ObjectsTopic: 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
Dissemination level: Public
Contractual delivery date: 31/10/2021
Actual delivery date: 15/10/2021

Project website address: https://remodel-project.eu

1 Introduction

Deformable Linear Objects (DLOs) is a class of objects that is characterized by two main features: (I) deformability, which refers to the fact that the object is not a rigid body and its geometry can change, (i)linearity, which stands for the fact that the object is elongated and the ratio of its length to its width is relatively big [1]. Objects of this type are ubiquitous both in everyday life and in the industry, where one can find ropes, cables, pipes, sutures, etc. While the manipulation of the rigid objects is already solved for many interesting objects, manipulating DLOs is a complex and vital task, which has been in the scope of the researchers for over three decades [2]. The interest in this topic has grown over the last few years, as the automatic wiring harness assembly is of crucial importance for car manufacturers [3], as well as automatic completing surgical sutures can help surgeons [4].The main issue in DLO manipulation is perceiving the shape of the DLO, which is crucial for closing the feedback loop and modeling it to predict its future behaviors or learn some behaviors using simulation. These two factors make the DLO manipulation problem particularly hard and interesting. Throughout the years, there were multiple attempts made to solve these problems and, as a consequence, to create a solution for DLO manipulation. To perform DLO tracking, one has to propose an algorithm that transforms the data from sensors into the chosen low-dimensional representation, serving as a DLO state. While there are attempts to use data from tactile sensors [5], the most successful way to perceive the DLO shape is to use vision and depth sensors. One of the most straightforward approaches to DLO shape tracking is to use the fiducial markers located along the DLO and track them [6] or use them to estimate the DLO shape[7]. A more sophisticated approach, which utilizes the walks on the region adjacency graphs built with super-pixels, was presented in [8]. These methods focus on tracking by directly using the measurements as the DLO state. However, the most common approaches utilize a model of the DLO and use images or point clouds as measurements to iteratively modify its parameters and track the object deformation. In [9],a Structure Preserved Registration algorithm with the object represented as a Mixture of Gaussians was used. Authors of [10] performed DLO tracking using Recursive Bayesian Estimator on Spatial Distribution Model, built with the Bezier curve and the chain of rectangles. In [11] authors proposed an approach that extends the Gaussian Mixture Model with Expectation-Maximization with a Coherent Point Drift method and additional constraints and regularization terms that encourage the estimation result to be physically plausible. Due to the iterative, and often probabilistic character of the model updates, these methods usually have problems tracking rapidly deforming objects, requiring an appropriate model and accurate initialization. To mitigate the issue of the slow initialization, authors of [12] used the Euclidean minimum spanning tree and the breadth-first search method to perform fast initialization and then used EM for updating the B-spline model of the DLO. However, it still takes hundreds of milliseconds to obtain the DLO shape estimate. While valid and important to perform the tracking, these models usually do not incorporate the cable's physics,so they cannot solve the second problem – prediction of the future evolution of the DLO. A step towards physics-based models was made in [13], where a modified EM algorithm is used to update the predefined DLO model based on the registered deformations and simulation in the physics engine, or in [14], where the FEM methods were used to track the deformation of the predefined model. While being able to predict the beavior of the deformable object when subjected to some external stimulation, these models are rather only local and valuable to aid the tracking process but not to predict the long-term evolution of the state. Thus,to enable planning, some researchers started to create DLOs models that are firmly rooted in the physics of these objects [15, 16]. However, because of the long time needed to perform a simulation, these methods are hard to use in practice to solve the DLO shape planning problem.

Considering all these advancements in the DLO shape tracking and problems with modeling the DLOs physics, most DLO manipulation methods avoid using sophisticated models and rely mainly on the frequently updated linear ones, or utilize some learning methods to model DLO behavior using data-based approaches. The second group, learning-based approaches, prevail in the case of rope manipulation. In [17] an imitation learning approach was used to mimic the human performing some manipulation sequence for the rope lying on the table. Moreover, instead of relying on some model derived from rope physics, authors proposed to learn pixel-level inverse dynamics of the rope directly from images using self-supervised learning. Self-supervision can also be used to learn how to estimate the state of the rope from the image and then use it to perform model predictive control, as it was done in [18]. While, authors of [19] used Casual InfoGAN to generate imagined plans, represented as a sequence of images of plausible rope configurations, that connects the initial and desired scene image. A different approach to rope manipulation was described in [20], where learning the model of the rope, was replaced with a system that utilized the coherent point drift method to update the rope model frequently. Moreover, instead of learning how to perform some rope manipulation movements, authors created a predefined set of movement primitives that transformed the rope from one state to another and stacked them in sequences to perform some predefined tasks. A more general approach to the manipulation of DLOs, which also utilize the feedback from the DLO online shape estimation, was presented in [21, 22, 23]. These approaches try to approximate the behavior of the cable using a simple locally valid linear model of the cable kinematics updated online using newly acquired DLO shape measurements and robot configurations. In [23] an autoencoder neural network was used to extract the shape of the cable from images, and gradient-based updates of the local model were made. In turn, both in [21] and [22] a simple centerline extraction methods were applied to extract the DLO shape and updates were made using gradient-free methods such as moving window least-squares [21], recursive least squares, extended and unscented Kalman filter [22]. Unfortunately, all these methods require some algorithm to determine the order of the shape representation coordinates, and the algorithms used in those papers could have problems with handlin more complicated but not less serious situations when part of the cable is occluded or cable is bent in such a way that it intersects with itself. In this report, we present an improved approach to bi-manual cable manipulation. Using visual feedback and a local model of cable kinematics updated using recursive least squares method, we are able to generate manipulators movement that transforms the handled cable into the desired shape. The proposed solution can handle a much broader range of shapes in more challenging conditions (occlusions) compared to the State-of-the-Art algorithms.

2 Considered problem

The problem we consider in the demonstrator and this report is how to manipulate the cable S using two robotic manipulators to morph it into some desired shape S_r. We are assuming that the cable is rigidly grasped by the two robotic arms, at its ends, and throughout the manipulation remains in the camera's field of view, which is the only source of information about the cable shape and can be subjected to occlusions. The manipulation is performed at low speeds, using the incremental position motions, such that the behavior of the cable is not dependent form the transient states and cable dynamics and is determined by the equilibrium of its elastic and potential energy. The error of the achieved object's shape e is measured using the distance between sequence of points on the cable S and points on the reference curve S_r and is defined by S_r

$$e = \sum_{i=0}^{N} \left| S\left(\frac{i}{N}\right) - S_r\left(\frac{i}{N}\right) \right|, \tag{1}$$

where $S$and S_r are defined as 2D B-spline curves parameterized with a normalized curve length, and $|\cdot|$is a euclidean norm.

Visually the problem is presented in Figure 1, where in the first image there is a red cable in some initial configuration S_0 and the desired shape denoted with the blue cable shadow S_r, while on the second one is a result of successful manipulation, where cable final shape S_f is the same as the desired shape S_r.

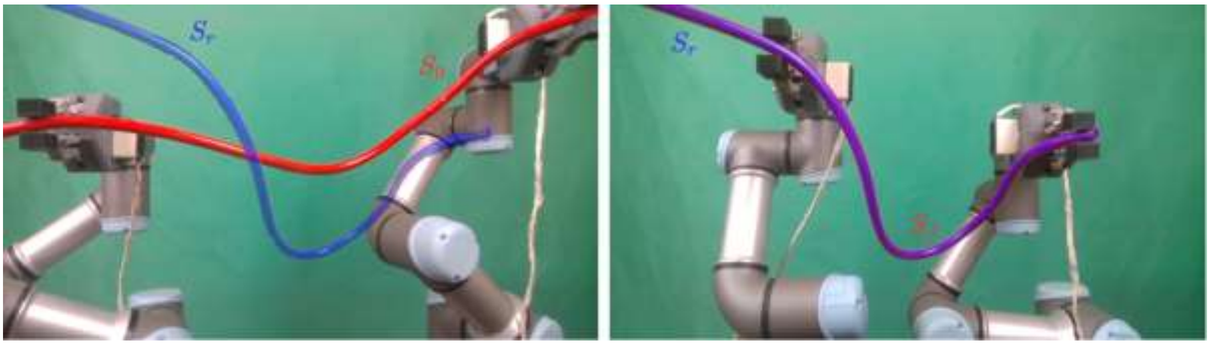

Figure 1: In this report, we consider the problem of deforming the cable from the initial shape S_0 (left image) to the desired shape S_r (right image) using two manipulators.

3 Proposed method

To solve the problem introduced above, we utilize the fast cable tracking system developed in the WP4, which we describe here briefly. Having the frequent measurements of the cable shape S, we can use methods similar to the ones introduced in [22] to regress the local linear model of the cable deformation and use it to generate moves of two robotic arms that leads the cable towards the goal shape S_r.

The general scheme of the proposed solution is presented in Figure 2. While manipulators are moving the cable, visual feedback pipeline segments the cable from the image, recognizes the cable's shape, and approximates it with a B-spline curve. Using this information controller can update the model of the cable deformation and, using a reference shape S_r, produces a positional update that transforms the cable towards the desired shape. In the following subsections, we will describe the parts mentioned above of the bi-manual cable manipulation algorithm.

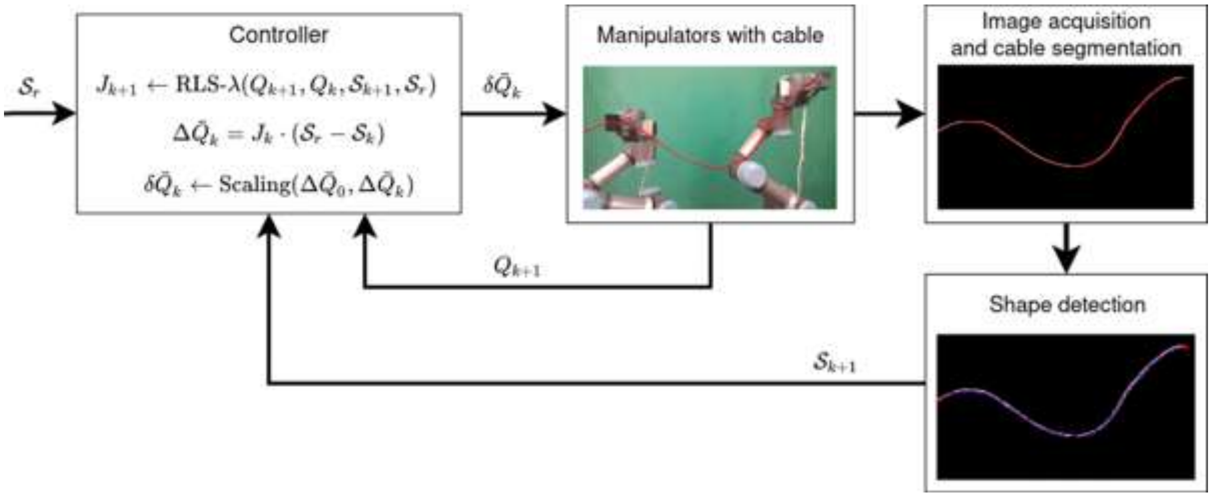

Figure 2: General scheme of the proposed solution for bi-manual robotic cable manipulation.

Visual feedback

An essential part of the proposed approach to the bi-manual cable manipulation, which does not use an accurate physics-based cable model, is the fast visual feedback that enables us to immediately track the behavior and update a simple local model of the cable, and perform visual-servoing like manipulation.

At the very first stage, the cable is segmented out from the image. Even though this can be done using different algorithms i.e., neural network-based approaches to object segmentation, in this demonstrator, for the sake of simplicity, we segmented the cable based on its color.

In the second step of the image processing skeletonization is applied to the segmented image to obtain simplified information about the shape of the segmented cable or its parts in case of the occlusions. Next, to be able to handle more complicated cable configurations, such as self-intersections, we remove from the skeletonized image the pixels with more than two neighbors, which results in splitting the cable into parts at every point where self-occlusion or some accidental branching (which may be caused by the skeletonization procedure) occurs.

Then, the skeletonized segments are connected to each other to form a single cable (sequence of segments), taking into account the distance between segments ends and their orientations.

Then finally, we fit a B-spline curve into the formed sequence of segments achieving the resultant shape of the cable S.

Cable deformation model

In general, modeling the cable dynamics or kinematics is a challenging task, which is still not solved on a satisfactory level, as the models from the literature require intensive computations to simulate the cable behavior [15,16]. Therefore, one has to give up some of the properties of those models, like globality or accuracy of the model. In our work, we decided to forfeit the globality of the model and use some local model, which can be rapidly updated given new observations and deviations from the predicted behavior.

Since we consider physical objects like cables, we can assume that small movements of the robot's TCPs $\Delta Q = Q_k - Q_{k-1}$ should lead to the slight deviations $\Delta S = S_k - S_{k-1}$ in terms of the cable shape S. Thus, we can define a local deformation model

$$\Delta S = J \Delta Q \tag{2}$$

where $J \in \mathbb{R}^{N \times q}$ is the jacobian-like matrix that represents the locally valid linear dependencies between the change of the robots TCP configurations and cable shape, where N is the size of shape representation, while q is the shape of TCPs configuration vector Q.

In this work, since the visual feedback is provided with a single camera, we will be operating on the 2D plane perpendicular to the camera's optical axis. Therefore, the TCPs configuration can be defined by

$$Q = \begin{bmatrix} x_I \\ y_I \\ \theta_I \\ x_{II} \\ y_{II} \\ \theta_{II} \end{bmatrix} \in \mathbb{R}^6, \tag{3}$$

where x_I and x_{II} denote the position in the task space of the robot's TCPs in the vertical axis of the image plane for the first and second arm, respectively. Similarly, y_I, y_{II} describe the position in the horizontal axis, while θ_I, θ_{II} the rotation around the camera optical axis. As one can see, the other rotations and movement along the optical axis are not considered, as they could cause deformations that are not distinguishable using a single camera. While this seems to be some limitation of the proposed method, the usage of more than one camera, with optical axes pointing in different directions, can still lead to 3D bimanual manipulation. Ensuring that each camera provides feedback only about the deformations clearly visible in the images it acquires, thus they constitute evidence to perform certain movements, leads to a reliable and safe manipulation.

Similarly, the shape of the cable is defined in an image space -- on a 2D plane. From the visual feedback, we obtain a B-spline curve S, whose shape corresponds with the actual shape of the cable. To use this for local cable modeling, we discretize this curve and obtain a sequence $\bar{S}$ of N equally distant points of the curve S, which can be defined by

$$\bar{S} = \left[S(0) \quad S\left(\frac{1}{N-1}\right) \quad \dots \quad S(1) \right] \in \mathbb{R}^{N\times 2}, \tag{4}$$

where $S(\cdot)$ is a point on the 2D image plane.

Cable manipulation

Having the model defined, we introduce three important procedures that lead to the bimanual cable manipulation: model identification, model update, and movement prediction.

To manipulate the cable based on the local linear model, one has to first identify its parameters before starting using it for the movement predictions. To do so, both arms are performing M specific movements during which the changes of configuration Q and cable shape $\bar{S}$ are gathered and stored. As a result, we obtain the matrix of configurations changes $\Delta Q_{init} \in \mathbb{R}^{6\times M}$ and matrix of cable shape changes $\Delta S_{init} \in \mathbb{R}^{N\times M}$. Having these measurements, one can easily regress the value of J_0 using the least-squares method

$$J_0 = \Delta S_{init} \Delta Q_{init}^{\dagger}. \tag{5}$$

Next, using the actual value of the J_k matrix, one can predict the movement of the arms $\Delta\bar{Q}$, which according to the identified local model, will minimize the difference between the actual S_k and the desired cable shape S_r, for $k = 0,1,2, \dots$. To do so, one has to calculate the difference $\Delta S_k = S_r - S_k$ and transform it using the local cable model into $\Delta\bar{Q}$ using the following formula

$$\Delta\bar{Q}_k = J_k^{\dagger} \Delta S_k \tag{6}$$

However, this transformation because of the model locality can generally be invalid and lead far from the desired cable shape. Therefore, to take the model locality into account, we do not apply the $\Delta\bar{Q}_k$ movement directly but scale it to obtain some short local move. This ensures that we do not enter areas where our local model is no longer useful (it cannot predict the cable behavior within some small neighborhood of the new configuration).

To achieve the desired shape of DLO, a predicted TCPs transformation described as

$$\Delta\bar{Q}_k = \begin{bmatrix} \Delta x_{I_k} \\ \Delta y_{I_k} \\ \Delta\theta_{I_k} \\ \Delta x_{II_k} \\ \Delta y_{II_k} \\ \Delta\theta_{II_k} \end{bmatrix}, \tag{7}$$

is taken under a scaling procedure required to perform movement sequence using partial transformation

$$\delta\bar{Q}_k = \begin{bmatrix} \delta x_{I_k} \\ \delta y_{I_k} \\ \delta\theta_{I_k} \\ \delta x_{II_k} \\ \delta y_{II_k} \\ \delta\theta_{II_k} \end{bmatrix}. \tag{8}$$

With an initial $\delta\bar{Q}_0$the TCPs translation δd_0 and rotation $\delta\theta_0$are associated. The transformation of TCPs have to be calculated separately

$$\begin{gathered} \delta x_{I_k} = \frac{\Delta d_{I_k}\delta d_k cos(\alpha_{I_k})}{\Delta d_k}, \\ \delta y_{I_k} = \frac{\Delta d_{I_k}\delta d_k sin(\alpha_{I_k})}{\Delta d_k}, \\ \delta\theta_{I_k} = \frac{\Delta\theta_{I_k}\delta\theta_k}{\Delta\theta_k}, \\ \delta x_{II_k} = \frac{\Delta d_{II_k}\delta d_k cos(\alpha_{II_k})}{\Delta d_k}, \\ \delta y_{II_k} = \frac{\Delta d_{II_k} d_k sin(\alpha_{II_k})}{\Delta d_k}, \\ \delta\theta_{II_k} = \frac{\Delta\theta_{II_k}\delta\theta_k}{\Delta\theta_k}, \end{gathered} \tag{9}$$

where

$$
\begin{aligned}
\Delta d_{I_k} &= \sqrt{\Delta x_{I_k}^2 + \Delta y_{I_k}^2},\\
\Delta d_{II_k} &= \sqrt{\Delta x_{II_k}^2 + \Delta y_{II_k}^2},\\
\Delta d_k &= \Delta d_{I_k} + \Delta d_{II_k},\\
\alpha_{Ik} &= atan2(\Delta y_{I_k}, \Delta x_{I_k}),\\
\alpha_{IIk} &= atan2(\Delta y_{II_k}, \Delta x_{II_k}),\\
\Delta\theta_k &= \Delta\theta_{I_k} + \Delta\theta_{II_k}.
\end{aligned}
\tag{10}
$$

By processing consecutive movements towards desired DLO shape, the $\delta\bar{Q}_k$ values change and are using as a scaling factor for the subsequent translation and rotation

$$
\begin{aligned}
\delta d_{k+1} &= \frac{\Delta d_k \delta d_0}{\Delta d_0},\\
\delta\theta_{k+1} &= \frac{\Delta\theta_k \delta\theta_0}{\Delta\theta_0},
\end{aligned}
\tag{11}
$$

where

$$
\begin{aligned}
\Delta d_k &= \Delta d_{I_k} + \Delta d_{II_k},\\
\Delta\theta_k &= \Delta\theta_{I_k} + \Delta\theta_{II_k}.
\end{aligned}
\tag{12}
$$

After every move is performed, one can analyze the model's accuracy and update it accordingly using the information from the last movement. Check of the model prediction can be made with the following formula

$$
\Delta S_{pred} = J_k \delta Q_k, \tag{13}
$$

then the difference between ΔS_{pred} and measured ΔS_k will inform us about the prediction error. This difference is an essential part of the model update, which we made using the recursive least squares method with $\lambda < 1$. The formula for updating the model is the following

$$
J_{k+1} = J_k + \frac{(\Delta S_{k+1} - \Delta S_{pred})\delta Q_k^T U_k}{\lambda + \delta Q_k^T U_k \delta Q_k}, \tag{14}
$$

where $U_k \in \mathbb{R}^{6\times 6}$is a matrix that can be defined with the following recursive algorithm

$$
U_{k+1} = \frac{1}{\lambda}\left(U_k - \frac{U_k \delta Q_k \delta Q_k^T U_k}{\lambda + \delta Q_k^T U_k \delta Q_k}\right), \tag{15}
$$

where $U_0 = \lambda_0^{-1} I$assuming that $I \in \mathbb{R}^{6\times 6}$is an identity matrix.

Thanks to this algorithm it is possible to update the model for every move robots make, thus obtaining a cable deformation model $J_k$up to date and valid for the local neighborhood of the actual cable configuration.

Frequent updates should allow for keeping local estimation of the model accurate enough to perform successful manipulation. However, because of measurement noises and the complicated behavior of the cable, while being bend, the model may change significantly between iterations in some situations. We expect that at the very beginning when the shape error e is high, the changes of the jacobian will be significant, whereas, at the neighborhood of the desired shape, we expect that the cable deformation model will vary only a bit. Therefore, we made the value of the forgetting factor λdependent on the ratio of actual e_k and initial e_0 cable shape errors and describe its evolution with the following equation

$$\lambda_k = 1 - \frac{e_k \lambda_0}{e_0}. \tag{16}$$

Algorithm

To sum up, the complete algorithm we proposed for bimanual cable manipulation can be written as follows:

1. Perform predefined movements with both arms and store the resultant differences in robots configurations ΔQ_{init} and cable shapes ΔS_{init}
2. Calculate the initial jacobian J_0using (5)
3. Start the manipulation sequence and set $k = 0$
4. Using the actual jacobian J_k and the difference between actual and desired cable shape ΔS_k calculate the predicted change of robots configurations $\Delta \bar{Q}_k$ that, according to the deformation model J_k is needed to deform the cable to the desired shape S_r using (6)
5. Apply the scaling procedure (8-12) to $\Delta \bar{Q}_k$ to obtain the small movement $\delta \bar{Q}$ towards the goal
6. Perform the movements of robots according to the $\delta \bar{Q}$
7. Measure the shape of the deformed cable S_{k+1}
8. Calculate the cable shape ΔS_{pred} after $\delta \bar{Q}$ move, predicted with the use of J_k and (13)
9. Calculate J_{k+1} based on the cable shape prediction error $\Delta S_{k+1} - \Delta S_{pred}$ using RLS-λ algorithm (14, 15)
10. Update the λ using (16)
11. Calculate the distance between actual and desired cable shape e_{k+1} using (1)
12. Compare the actual $e_{k+1}$with the W previous ones, and if there was no improvement since W iterations, move the robots to the best-achieved configurations and stop the algorithm ($W$is a constant that can be chosen by the user).

13. Increment k and go to point 4.

4 ROS node description

ROS2 was used to deliver the described functionality (Figure 3). This allowed us to use a previously implemented algorithm for cable tracking, which uses the input image to generate a sequence of points describing the shape of the B-spline. A change within the B-spline $\Delta S = S_k - S_{k-1}$is associated with each transformation δQ until the defined sequence of moves is completed. From such a data set, the model J_0 necessary to predict movements toward the target is estimated. After each predicted movement δQ_k, the model is optimized for the next iteration using the resulting absolute error e_k between points of B-spline expected and B-spline achieved. This process is repeated until the algorithm is unable to minimize e_g (which is an absolute error between points of B-spline goal and B-spline achieved) in a certain consecutive number of movements and defined as a condition that ends the algorithm.

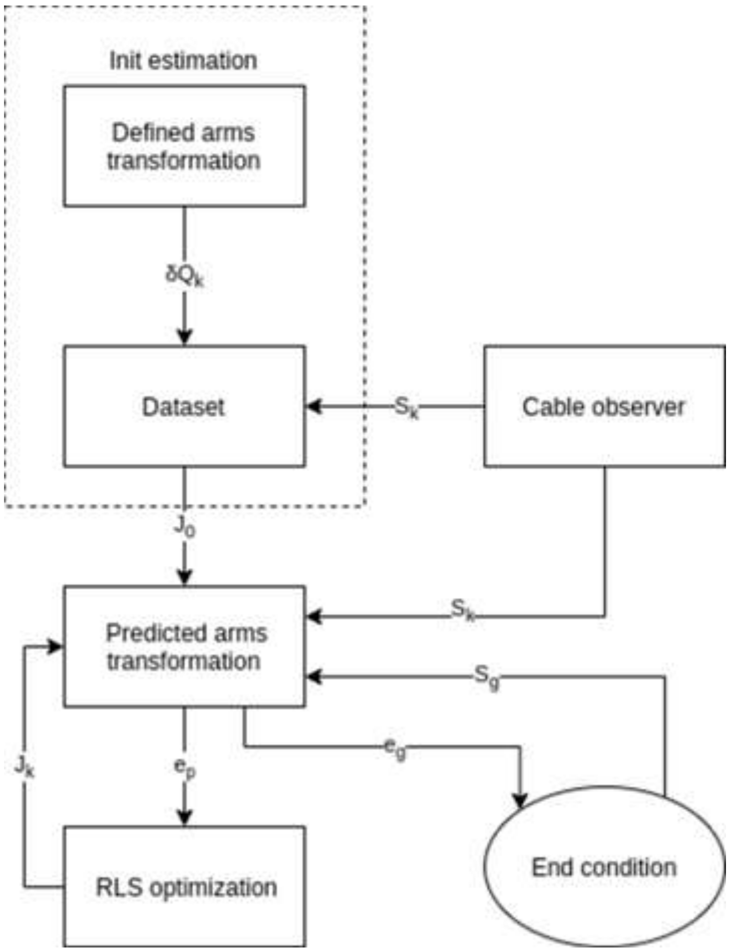

Figure 3: General scheme of implemented algorithm.

5 Experiments

We experimentally evaluated the proposed bi-manual cable manipulation algorithm in a laboratory at Poznan University of Technology. Two UR3 arms, equipped with our own designed grippers were manipulating the cable in front of the RealSense D435 camera, that was providing the visual feedback. To show the robustness of the proposed algorithm, we performed the manipulation of a few different DLOs (different lengths, weights, stiffness). Also, we tested our approach in demanding situations, including occlusions and self-intersections. The results of the performed experiments are presented in the demonstrator movie[1] and the figures below.

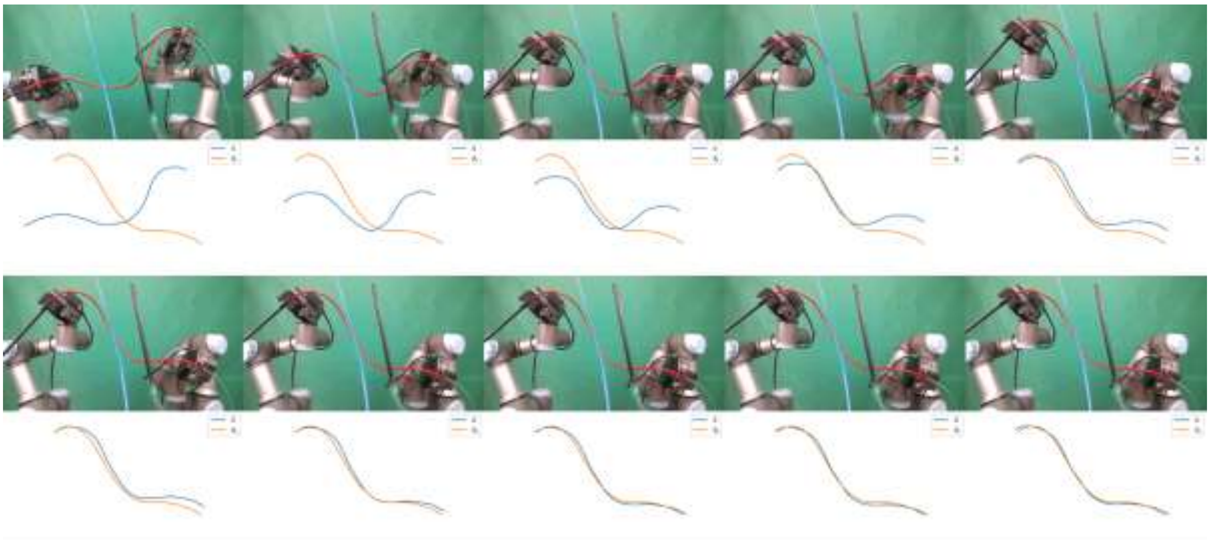

Figure 4: Manipulation sequence for a thin cable subjected to some occlusions

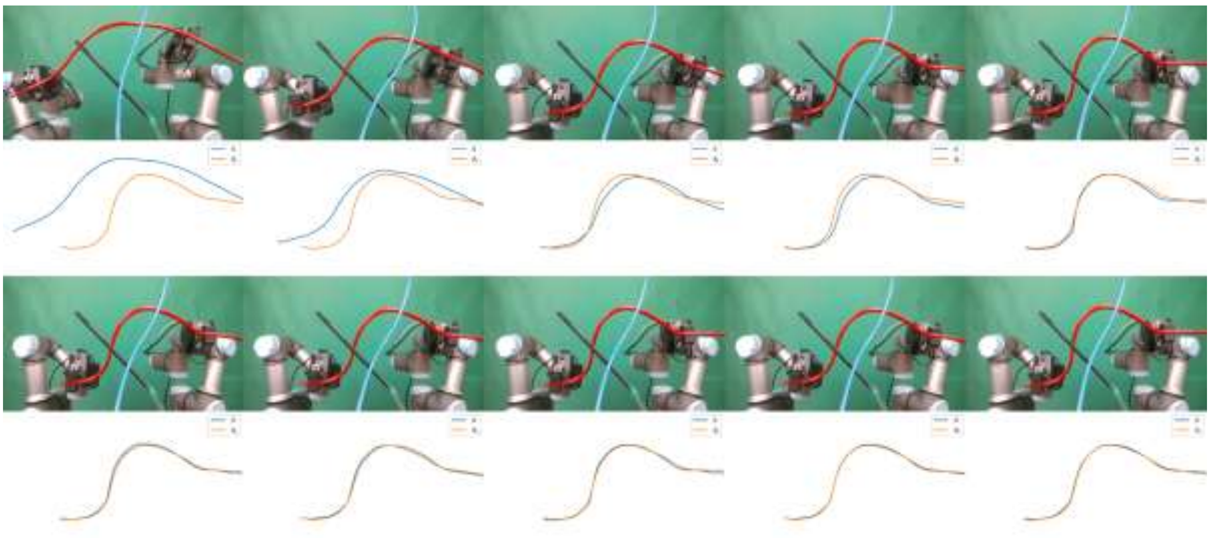

Figure 5: Manipulation sequence for a thick cable subjected to some occlusions

[1] http://intranet.remodel-project.eu/share/s/hv33dCNeSI2vaedvidu7lg

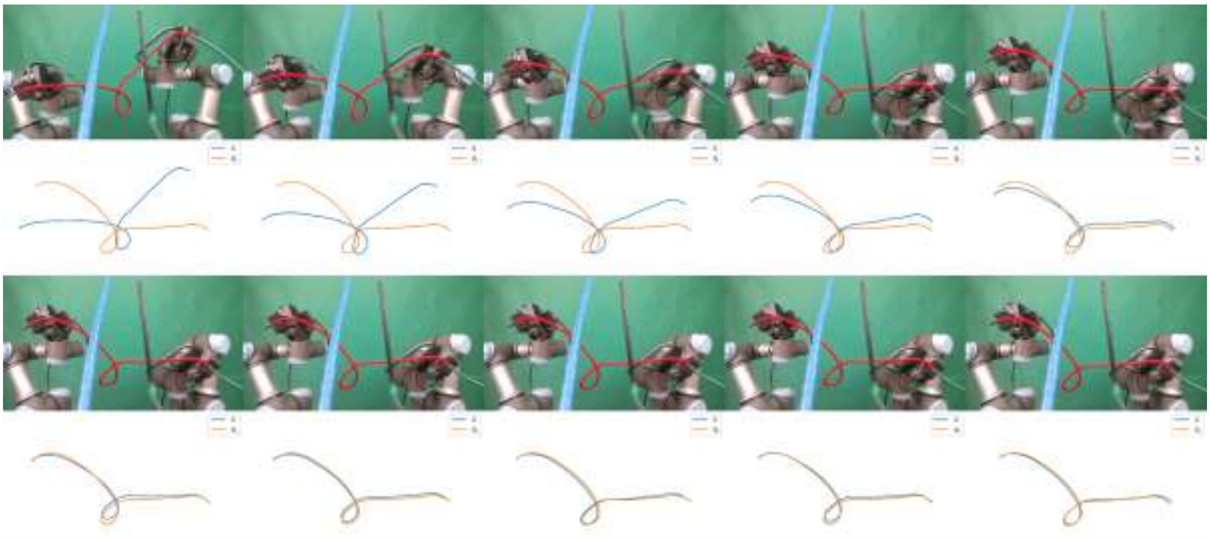

Figure 6: Manipulation sequence for a medium thickness cable with a loop subjected to larger occlusions

6 Conclusions

To sum up, we have prepared a ROS node[2] that enables deforming a cable into the desired shape using two robotic arms holding it and visual feedback provided by a camera. The presented solution works on several different cables in demanding setups in our laboratory. However, it can be easily adapted to work on different hardware.

[2] https://dei-gitlab.dei.unibo.it/aszymko/putarms_jacobian/-/tree/2d

7 References

[1] J. Sanchez, J.-A. Corrales, B.-C. Bouzgarrou, Y. Mezouar, Robotic manipulation and sensing of de-formable objects in domestic and industrial applications: a survey, The International Journal of RoboticsResearch 37 (7) (2018) 688–716. doi:10.1177/0278364918779698.URLhttps://doi.org/10.1177/02783649187796986

[2] M. Inaba, H. Inoue, Rope handling by a robot with visual feedback, Advanced Robotics 2 (1) (1987)39–54. arXiv:https://doi.org/10.1163/156855387X00057, doi:10.1163/156855387X00057.URLhttps://doi.org/10.1163/156855387X00057

[3] P. Kicki, M. Bednarek, P. Lembicz, G. Mierzwiak, A. Szymko, M. Kraft, K. Walas, Tell me, what do yousee?—interpretable classification of wiring harness branches with deep neural networks, Sensors 21 (13)(2021). doi:10.3390/s21134327.URLhttps://www.mdpi.com/1424-8220/21/13/4327

[4] S. Sen, A. Garg, D. V. Gealy, S. McKinley, Y. Jen, K. Goldberg, Automating multi-throw mul-tilateral surgical suturing with a mechanical needle guide and se-quential convex optimization, in:2016 IEEE International Conference on Robotics and Automation (ICRA), 2016, pp. 4178–4185.doi:10.1109/ICRA.2016.7487611.

[5] J. Sanchez, C. M. Mateo, J. A. Corrales, B. Bouzgarrou, Y. Mezouar, Online shape esti-mation based ontactile sensing and deformation modeling for robot manipulation, in: 2018 IEEE/RSJ International Con-ference on Intelligent Robots and Systems (IROS), 2018, pp. 504–511. doi:10.1109/IROS.2018.8594314.

[6] Y. Lai, J. Poon, G. Paul, H. Han, T. Matsubara, Probabilistic pose estimation of deformable linearobjects, in: 2018 IEEE 14th International Conference on Automation Science and Engineering (CASE),2018, pp. 471–476. doi:10.1109/COASE.2018.8560497.

[7] A. Borum, D. Matthews, T. Bretl, State estimation and tracking of deforming pla-nar elastic rods,in: 2014 IEEE International Conference on Robotics and Automa-tion (ICRA), 2014, pp. 4127–4132.doi:10.1109/ICRA.2014.6907459.

[8] D. De Gregorio, G. Palli, L. Di Stefano, Let's take a walk on superpixels graphs: Deformable linearobjects segmentation and model estimation, in: C. V. Jawahar, H. Li, G. Mori, K. Schindler (Eds.),Computer Vision – ACCV 2018, Springer International Publishing, Cham, 2019, pp. 662–677.

[9] T. Tang, M. Tomizuka, Track deformable objects from point clouds with structure pre-served registration, The International Journal of Robotics Research 0 (0) (0) 0278364919841431.doi:10.1177/0278364919841431.URLhttps://doi.org/10.1177/0278364919841431

[10] A. Zea, F. Faion, U. D. Hanebeck, Tracking elongated extended objects using splines, in: 2016 19thInternational Conference on Information Fusion (FUSION), 2016, pp. 612–619.

[11] Y. Wang, D. McConachie, D. Berenson, Tracking partially-occluded deformable objects while enforcinggeometric constraints (2020). arXiv:2011.00627.

[12] G. Yao, R. Saltus, A. P. Dani, Shape estimation for elongated deformable object using b-spline chainedmultiple random matrices model, International Journal of Intelligent Robotics and Applications 4 (4)(2020) 429–440. doi:10.1007/s41315-020-00149-w.URLhttps://doi.org/10.1007/s41315-020-00149-w

[13] J. Schulman, A. Lee, J. Ho, P. Abbeel, Tracking deformable objects with point clouds,in:2013 IEEE International Conference on Robotics and Automation, 2013, pp. 1130–1137.doi:10.1109/ICRA.2013.6630714.

[14] A. Petit, V. Lippiello, B. Siciliano, Real-time tracking of 3d elastic objects with an rgb-d sensor, in: 2015IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 2015, pp. 3914–3921.doi:10.1109/IROS.2015.7353928.

[15] G. Palli, Model-based manipulation of deformable linear objects by multivariate dynamic splines,in: 2020 IEEE Conference on Industrial Cyberphysical Systems (ICPS), Vol. 1, 2020, pp. 520–525.doi:10.1109/ICPS48405.2020.9274730.7

[16] J. Spillmann, M. Teschner, Corde: Cosserat rod elements for the dynamic simulation of one-dimensionalelastic objects, in: Proceedings of the 2007 ACM SIGGRAPH/Eurographics Symposium on ComputerAnimation, SCA ’07, Eurographics Association, Goslar, DEU, 2007, p. 63–72.

[17] A. Nair, D. Chen, P. Agrawal, P. Isola, P. Abbeel, J. Malik, S. Levine, Combining self-supervised learningand imitation for vision-based rope manipulation, CoRR abs/1703.02018 (2017). arXiv:1703.02018.URLhttp://arxiv.org/abs/1703.02018

[18] M. Yan, Y. Zhu, N. Jin, J. Bohg, Self-supervised learning of state estimation for manipulating deformablelinear objects, CoRR abs/1911.06283 (2019). arXiv:1911.06283.URLhttp://arxiv.org/abs/1911.06283

[19] A. Wang, T. Kurutach, K. Liu, P. Abbeel, A. Tamar, Learning robotic manipulation through visualplanning and acting, CoRR abs/1905.04411 (2019). arXiv:1905.04411.URLhttp://arxiv.org/abs/1905.04411

[20] T. Tang, C. Wang, M. Tomizuka, A framework for manipulating deformable linear objectsby coherent point drift, IEEE Robotics and Automation Letters 3 (4) (2018) 3426–3433.doi:10.1109/LRA.2018.2852770.

[21] J. Zhu, B. Navarro, P. Fraisse, A. Crosnier, A. Cherubini, Dual-arm robotic manipulation of flexiblecables, in: 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 2018,pp. 479–484. doi:10.1109/IROS.2018.8593780.

[22] J. Qi, W. Ma, D. Navarro-Alarcon, H. Gao, G. Ma, Adaptive shape servoing of elastic rods us-ing parameterized regression features and auto-tuning motion controls, CoRR abs/2008.06896 (2020).arXiv:2008.06896.URLhttps://arxiv.org/abs/2008.06896

[23] J. Qi, G. Ma, D. Navarro-Alarcon, H. Zhang, Y. Lyu, Towards latent space based manipulation ofelastic rods using autoencoder models and robust centerline ex-tractions, CoRR abs/2101.07513 (2021).arXiv:2101.07513.URLhttps://arxiv.org/abs/2101.075138