

Alma Mater Studiorum Università di Bologna  
Archivio istituzionale della ricerca

Model-based Manipulation of Deformable Linear Objects by Multivariate Dynamic Splines

This is the final peer-reviewed author's accepted manuscript (postprint) of the following publication:

*Published Version:*

Palli G. (2020). Model-based Manipulation of Deformable Linear Objects by Multivariate Dynamic Splines. Institute of Electrical and Electronics Engineers Inc. [10.1109/ICPS48405.2020.9274730].

*Availability:*

This version is available at: <https://hdl.handle.net/11585/796312> since: 2021-04-08

*Published:*

DOI: <http://doi.org/10.1109/ICPS48405.2020.9274730>

*Terms of use:*

Some rights reserved. The terms and conditions for the reuse of this version of the manuscript are specified in the publishing policy. For all terms of use and more information see the publisher's website.

This item was downloaded from IRIS Università di Bologna (<https://cris.unibo.it/>).  
When citing, please refer to the published version.

(Article begins on next page)

This is the final peer-reviewed accepted manuscript of:

G. Palli, "Model-based Manipulation of Deformable Linear Objects by Multivariate Dynamic Splines," *2020 IEEE Conference on Industrial Cyberphysical Systems (ICPS)*, Tampere, Finland, 2020, pp. 520-525, doi: 10.1109/ICPS48405.2020.9274730.

The final published version is available online at:

<https://www.doi.org/10.1109/ICPS48405.2020.9274730>

Rights / License:

The terms and conditions for the reuse of this version of the manuscript are specified in the publishing policy. For all terms of use and more information see the publisher's website.

***When citing, please refer to the published version of the article as indicated above.***

# Model-based Manipulation of Deformable Linear Objects by Multivariate Dynamic Splines

Gianluca Palli

DEI - Department of Electrical, Electronic and Information Engineering  
University of Bologna, Viale Risorgimento 2, 40136 Bologna, Italy

**Abstract**—In this paper, the modelling and the simulation of a Deformable Linear Object (DLO) manipulation are reported. The main motivation of this study is to define a strategy to enable a robotic manipulator to predict in real time the shape a DLO will achieve during the execution of a manipulation action. To accomplish this target in a reasonable time, according to the possibility of adopting this solution in an industrial manufacturing system, an approximate but physically consistent model of the DLO is adopted considering the predominant plasticity of the object to be manipulated, as in the case of electric cable manipulation. The DLO manipulation model is based on multivariate dynamic splines solved iteratively in real-time to interpolate the DLO shape during the manipulation sequence. The systems assumes to be able to detect the initial configuration of the DLO at each iteration of the algorithm by means of a proper vision system. Preliminary simulation results are presented to show the effectiveness of the method.

**Index Terms**—Robotic Manipulation, Deformable Objects, Modeling and Simulation, Optimization, Nonlinear Systems.

## I. INTRODUCTION

The manipulation of deformable objects represents a relevant topic of interest for the implementation on novel manufacturing applications in several sectors, not only in food production [1] and textile manufacturing [2] but also in surgery [3], [4] and in the automotive, aerospace [5] and electromechanical industries [6], [7] in general, where many assembly operations involving this kind of objects are still executed manually due to the variability of initial configurations and unpredictable behaviors. A thoughtful survey on robotic manipulation of deformable objects in domestic and industrial applications can be found in [8].

In this scenario, a relevant subfield of broad industrial interest is represented by the manipulation of Deformable Linear Objects (DLOs), such as ropes, electric wires and cables, strings and so on. Several literature works addressed the modeling and the manipulation of this kind of objects for several purposes and several different models and strategies were developed. In [9] a geometrically consistent model of DLOs is developed and adopted to perform numerical simulations on the object motion under gravity and during the

interaction with the environment. In [10] a linearized spline-based DLO model called quasi dynamic splines is developed for the purpose of interaction simulation. A manipulation controller for an industrial robot inserting a cable on a tight hole without a priori knowledge of the cable parameters is proposed in [11]. The integration of robotic vision and tactile sensing for switchgear cabling tasks is reported in [12], and a vision-based rope manipulation combining self-supervised learning and imitation is proposed in [13].

The main objective of this paper is to enable a robot to shape a DLO according to a desired target shape. To this end, a mathematical but approximated model of the cable and of the manipulation task is implemented by means of multivariate dynamic splines. This model allows a fast and real-time evaluation of the manipulation action toward the desired goal. The definition of the model is defined as a trade-off between accuracy and execution time. This approach is justified by the fact that, in this scenario, several source of uncertainties are present, such as limited knowledge and variability of material parameters, nonlinear and hard-to-model effects like plasticity and friction, imprecise shape reconstruction, non-ideal manipulation conditions and so on. Therefore, the use of a precise but complex model will probably provide a better solution according to the theoretical model, but due to the model and parameter uncertainties this solution will be practically not more effective than an approximated one.

In this paper, a single-arm manipulation system equipped with a parallel gripper is assumed, and the investigation is based on the assumption that the robot can only grasp the DLO at a certain point and move point in another position on the same plane (the table plane). The DLO segmentation and model estimation from 2D images investigated in [14] will be exploited to provide the input model for the algorithm initialization after the execution of each manipulation primitive, i.e. grasping, repositioning of the cable at the grasping point and release.

The remaining part of this paper is structured as follows. In Sec. II summarizes the main aspects related to the mathematical model of the DLO, while Sec. III describes how the manipulation primitive can be interpolated over time. Preliminary simulation results are reported in Sec. IV. Finally, in Sec. V, conclusions about the presented work and

This work was supported by the European Commissions Horizon 2020 Framework Programme with the project REMODEL - Robotic technologies for the manipulation of complex deformable linear objects - under grant agreement No 870133.

Corresponding author: gianluca.palli@unibo.it

future developments are draft out.

## II. DYNAMIC MODEL OF DLOS

### A. DLO Configuration and Lagrange Equation

The dynamic model of a DLO can be effectively represented by means of a 3-rd order spline basis as a function of a free coordinate  $u$  that represents a position along the cable starting from an end point, where  $u = 0$ , to the opposite end where  $u = L$ , being  $L$  the length of the cable

$$q(u) = \sum_{i=1}^{n_u} b_i(u)q_i \quad (1)$$

where  $q(u) = (x(u), y(u), z(u), \theta(u)) = (r(u), \theta(u))$  is the 4-th dimensional configuration functional space of the cable, i.e. including three linear coordinates  $x, y, z$  of the DLO position at point  $u$  and the axial DLO twisting  $\theta$ ,  $b_i(u)$  is the  $i$ -th elements of the spline polynomial basis used to represent the DLO shape and  $q_i$  are  $n_u$  properly selected coefficients, usually called *control points*, used to properly interpolate the DLO shape through the  $b_i(u)$  function basis.

This mathematical model of DLO is very effective for a number of reasons. First, the computation of the shape spatial derivatives is straightforward, i.e.

$$q'(u) = \sum_{i=1}^{n_u} b'_i(u)q_i \quad (2)$$

$$q''(u) = \sum_{i=1}^{n_u} b''_i(u)q_i \quad (3)$$

and, most notably, can be represented through the same coefficients and simple-to-compute derivatives of the polynomial spline basis functions  $b_i(u)$ . Secondly, the proprieties of spline basis ensures minimization of the DLO model curvature [15], that represents the physical behavior of DLOs. Third, this model allows to represent a generic nonlinear function with sufficient smoothness properties as a linear combination of the nonlinear function basis  $b_i(u)$ , that depend only on the free variable  $u$ , by the linear coefficients  $q_i$ .

The dynamic model of the DLO can be defined as a function of the control points  $q_i$  by referring to the Lagrange equations of the system

$$\frac{d}{dt} \left( \frac{\partial T}{\partial \dot{q}_i} \right) = F_i - \frac{\partial U}{\partial q_i}, \quad \forall i \in \{1, \dots, n_u\} \quad (4)$$

where  $L$  is the length of the DLO,  $F_i$  is the resultant external force acting on the  $i$ -th control point,  $T$  is the overall kinetic energy of the system and  $U$  is the overall potential energy due to gravity, stretching, bending and torsional effects acting on the DLO.

### B. DLO Kinetic Energy and Inertial Force

The kinetic energy of the DLO is due to translation of the control points and rotation of the cross sections. The overall

kinetic energy can be represented as a function of the control points  $q_i$  as

$$T = \frac{1}{2} \int_0^L \frac{dq^T}{dt} J \frac{dq}{dt} ds, \quad J = \begin{bmatrix} \mu & 0 & 0 & 0 \\ 0 & \mu & 0 & 0 \\ 0 & 0 & \mu & 0 \\ 0 & 0 & 0 & I \end{bmatrix} \quad (5)$$

where  $ds = \|r'(u)\|du$  is the element displacement,  $J$  is the generalized density matrix of the DLO,  $\mu$  is the linear density and  $I$  is the polar moment of inertia.

Following the procedure described in [16], it is possible to write

$$\frac{d}{dt} \left( \frac{\partial T}{\partial \dot{q}_j} \right) = \sum_{j=1}^{n_u} J \int_0^L b_i(s)b_j(s)ds \frac{d^2 q_j}{dt^2} \quad (6)$$

By considering that  $\frac{d^2 q_j}{dt^2} = A_j$  is the acceleration of the  $j$ -th control point, the term  $J \int_0^L b_i(s)b_j(s)ds = M_{ij}$  can be considered the corresponding inertia term. Therefore, it is possible to write

$$\frac{d}{dt} \left( \frac{\partial T}{\partial \dot{q}_i} \right) = \sum_{j=1}^{n_u} M_{ij} \ddot{q}_j \quad (7)$$

and, extending this definition to the whole system, this allows to write the overall DLO inertial forces as  $M\ddot{q}$ , where  $M$  is the DLO inertia matrix and  $\ddot{q}$  is the vector of the control point accelerations.

### C. DLO Potential Energy and Elastic Forces

The potential energy  $U$  is composed by the gravity effect and by the DLO strain energy due to stretching, bending and torsion. While the derivation of the gravitational energy is quite straightforward given the previous discussion of the DLO inertia matrix, the definition of the strain energy plays a fundamental role in the DLO modelling. By defining the strain vector  $\epsilon = [\epsilon_s, \epsilon_b, \epsilon_t]$  including the stretching term  $\epsilon_s$ , the bending term  $\epsilon_b$  and the torsional term  $\epsilon_t$  such that

$$\begin{aligned} \epsilon_s &= 1 - \|r'\|, & \epsilon_t &= \theta' - \tau, & \epsilon_b &= \frac{\|C\|}{\|r'\|^3} \\ C &= r' \times r'', & \tau &= \frac{C^T r'''}{\|C\|^2} \end{aligned}$$

where  $r$  and  $\theta$  are the linear and the torsional component of the strain respectively.

It follows that the strain energy can be written as

$$U = \frac{1}{2} \int_0^L (\epsilon - \epsilon_0)^T H (\epsilon - \epsilon_0) ds = \frac{1}{2} \int_0^L \epsilon_e^T H \epsilon_e ds \quad (8)$$

where

$$H = \frac{D^2 \pi}{4} \begin{bmatrix} E & 0 & 0 \\ 0 & \frac{GD^2}{8} & 0 \\ 0 & 0 & \frac{ED^2}{16} \end{bmatrix}$$

is the element stiffness matrix,  $E$  and  $G$  are the Young modulus and the shear modulus of the material respectively,  $D$  is the DLO cross section diameter and  $\epsilon_0$  is the plastic strain, or *strain memory*, of the DLO, that allows to take into

account the plasticity of the material, and  $\epsilon_e = \epsilon - \epsilon_0$  is the residual strain.

The dynamics of the plastic strain can be represented as

$$\dot{\epsilon}_0 = \begin{cases} k_\epsilon(\epsilon_e - \epsilon_M), & \epsilon_e > \epsilon_M \\ 0, & -\epsilon_M \leq \epsilon_e \leq \epsilon_M \\ k_\epsilon(\epsilon_e + \epsilon_M), & \epsilon_e < -\epsilon_M \end{cases} \quad (9)$$

where  $\epsilon_M$  is a proper maximum strain threshold that generates a *memorization* of the actual strain when it is exceeded and  $k_\epsilon$  is positive a parameter used to adjust how fast the memorization process evolves. This strain memory is fundamental is the target application of this paper since the type of DLO of our interest, e.g. electric cables and plastic tubes, are characterized by dominant plasticity effect with respect to pure elasticity, in particular for what is related to the bending term.

The right side term of eq. (4) can be written as

$$P_i = -\frac{\partial U}{\partial q_i} = -\frac{1}{2} \int_0^L \frac{\partial \epsilon_e^T H \epsilon_e}{\partial q_i} ds = -\int_0^L \frac{\partial \epsilon}{\partial q_i} H \epsilon_e ds \quad (10)$$

representing the elastic forces due to the DLO deformation,

The strain derivative with respect to the  $i$ -th control point can be written as

$$\frac{\partial \epsilon}{\partial q_i} = \begin{bmatrix} \frac{r'^T b'_i}{\|r'\|} & 0 \\ \frac{\mathcal{T}_i^T}{\|\mathcal{C}\|^2} & \frac{b'_i}{2\|r'\|} \\ \frac{\|r'\|^2(\mathcal{C} \times \mathcal{P}_i)^T - 3\|\mathcal{C}\|^2 r'^T b_i}{\|\mathcal{C}\| \|r'\|^5} & 0 \end{bmatrix} \quad (11)$$

$$\mathcal{T}_i = \mathcal{C} b_i''' - \mathcal{P} \times r''' - 2\tau(\mathcal{C} \times \mathcal{P}), \quad \mathcal{P}_i = \frac{\partial r' \times r''}{\partial r_i}$$

where, with some abuse of notation, the first column of  $\frac{\partial \epsilon}{\partial q_i}$  is the  $3 \times 3$  matrix representing  $\frac{\partial \epsilon}{\partial r_i}$  and the second column is a vector representing  $\frac{\partial \epsilon}{\partial \theta_i}$ .

The time derivative of the strain can be then expressed by

$$\dot{\epsilon} = \sum_{i=1}^{n_u} \frac{\partial \epsilon}{\partial q_i} \frac{dq_i}{dt} = \Gamma^T \dot{q} \quad (12)$$

where  $\Gamma$  is the matrix collecting  $\frac{\partial \epsilon}{\partial q_i}$ ,  $\forall i$  and  $\dot{q}$  is the vector of the control point time derivative.

#### D. DLO Dynamic Model

By extending eqs. (4), (7) and (10) to all the control points, it is possible to write the overall DLO dynamic model as

$$M\ddot{q} + N\dot{q} = F + P \quad (13)$$

subject to eqs. (9) and (12), where the term  $N\dot{q}$  is introduced in order to take into account for DLO internal energy dissipation,  $F$  is the vector of all the external forces, gravity included, and  $P$  is the vector of all the elastic forces.

### III. PARAMETERIZATION OF THE MANIPULATION PRIMITIVE

In this section, the problem of defining the evolution of the DLO state, i.e. the evolution of its control points, when a manipulation action is performed on it is targeted. For a manipulation primitive, in this paper we mean the motion of a DLO point along a certain trajectory imposed by a robot grasping the DLO. This action will change the overall DLO state because of both the grasped point motion thanks to the internal elastic and inertial forces generated in the DLO by this action and by the external forces, such as gravity or contacts with obstacles. Given the dynamic model (13), an option could be to simulate the evolution of the DLO state over the manipulation action. The main problem on this approach is that, since we can not predict the resultant DLO state after the manipulation in advance, the evaluation of all the (potentially infinite) manipulation actions in order to define the best one will take too much time to be effective in a productive scenario. Moreover, this problem is worsened by the fact that very unlikely the goal DLO configuration will be achieved by a single manipulation action. This implies that multiple iterations of the manipulation evaluation problem are needed before reaching the task goal.

To solve this issue, we first need to take into account that an exact solution, such as the one that the numerical simulation of (13) will provide, is hard to be effective in a real scenario due to several causes, such as uncertainties in model parameters. Therefore, an approximated solution could be effective as well, or more effective if the time to compute it is reduced with respect to the exact solution. For this reason, a time-based spline interpolation of the DLO state, i.e. of the control points, over the manipulation primitive is introduced

$$q_i(t) = \sum_{j=1}^{n_t} b_j(t) q_{ij} \quad (14)$$

where  $n_t$  is the number of time-related control points, obtaining in this way a bivariate representation of the DLO configuration

$$q(u, t) = \sum_{i=1}^{n_u} \sum_{j=1}^{n_t} b_i(u) b_j(t) q_{ij} \quad (15)$$

where  $q_{ij}$  is a  $n_u \times n_t$  set of control points used to parameterize the problem. From eq. (15) it is straightforward to defined the time derivative of the DLO configuration

$$\dot{q}(u, t) = \sum_{i=1}^{n_u} \sum_{j=1}^{n_t} b_i(u) b'_j(t) q_{ij} \quad (16)$$

$$\ddot{q}(u, t) = \sum_{i=1}^{n_u} \sum_{j=1}^{n_t} b_i(u) b''_j(t) q_{ij} \quad (17)$$

This allows to significantly reduce the number of points to be computed in order to compute the DLO dynamics at the expense of providing an approximate (but effective) solution.

For the sake of compactness, we can rewrite the control points vector and its time derivative of as

$$q(t) = B_t(t)q_t, \quad \dot{q}(t) = B'_t(t)q_t, \quad \ddot{q}(t) = B''_t(t)q_t$$

where  $q_t$  is the vector stacking all the  $q_{ij}$  control points and  $B_t(t)$  is the block diagonal matrix having  $n_u$  lines generated by the line vector of coefficients  $[b_1(t) \cdots b_{n_t}(t)]$ . The same applies for  $B'_t(t)$  and  $B''_t(t)$  with the corresponding spline derivatives. Therefore, the DLO dynamic model (13) can be simply rewritten as

$$MB''_t q_t + NB'_t q_t = F + P \quad (18)$$

where the dependence from time of  $B_t$  and its derivatives is omitted for brevity. The left-hand side of eq. (18) is clearly linear in  $q_t$ , but since  $P$  depends on  $q$ , i.e.  $P = P(q)$ , the overall system is nonlinear and can not be solved efficiently in this form.

#### A. DLO Model Linearization

In order to solve eq. (18), a solution could be to perform numerical integration in a discrete-time fashion by evaluating the solution of the system at time  $t_{k+1}$  given the solution at time  $t_k$ , i.e.

$$MB''_t q_{k+1} + NB'_t q_{k+1} = F + P(B_t q_{k+1}) \quad (19)$$

where  $q_{k+1}$  and  $q_k$  are the values of  $q_t$  at time  $t_{k+1}$  and  $t_k$  respectively. This nonlinear model, besides being absolutely effective for DLO simulation, is computationally too expensive for an online evaluation of optimal manipulation strategies. However, a first step toward the simplification of this model is to rewrite the elastic energy term  $P(q_{k+1})$  in a linearized way

$$MB''_t q_{k+1} + NB'_t q_{k+1} + KB_t \Delta q_t = F + P(B_t q_k)$$

where  $K$  is the Hessian of the elastic energy  $U$  (which definition is omitted for brevity) and  $\Delta q_t = q_{k+1} - q_k$ . It follows that the linearized model of the DLO can be written as

$$\begin{aligned} MB''_t q_{k+1} + NB'_t q_{k+1} + KB_t q_{k+1} &= \\ &= F + P(B_t q_k) + KB_t q_k \end{aligned} \quad (20)$$

where now a clear separation between the values of  $q_t$  at time  $t_k$  and  $t_{k+1}$  is obtained. This model is suitable to be solved by an iterative algorithm, such as the Newton-Raphson or the conjugate gradient methods. Note that this model can be also used to find the DLO equilibrium by removing the terms  $MB''_t + NB'_t$ .

In particular, by collecting  $q_{k+1}$  the following linear system is obtained

$$(MB''_t + NB'_t + KB_t)q_{k+1} = F + P(B_t q_k) + KB_t q_k$$

or, in a more compact way

$$\Lambda q_{k+1} = \Phi(F, q_k) \quad (21)$$

$$\Lambda = MB''_t + NB'_t + KB_t$$

$$\Phi(F, q_k) = F + P(B_t q_k) + KB_t q_k$$

where  $\Lambda$  is a symmetric positive definite regression matrix, i.e. it is always invertible, assuming that the time-related spline basis  $B_t$  is properly defined, and  $\Phi(F, q_k)$  is a residual vector that can be computed without any problem at each iteration step.

#### B. Iterative Solution of the DLO Linearized Model

To illustrate the iterative solution scheme, let us considering the simpler case of computing the control points, or coefficients, describing the time evolution of the DLO configuration over a certain time period given the external forces  $F$ .

The algorithm is initialized by setting the control points at time step  $k = 0$ , let us call them  $q_0$ , equal to the actual DLO configuration. As mentioned in the introduction, we assume at this stage that a vision system, such as the one reported in [14], can be used to detect the DLO and provide a spline-based representation of its shape ( $q_0$  as it was named previously). Moreover, since we have no initial knowledge of the system input, i.e. the external forces  $F$ , we assume the DLO will maintain its shape along the time, i.e. all the time-related control points are set equal to the initial DLO configuration  $q_0$ . Also the internal residual strain  $\epsilon_0$  is set to the one given by the initial DLO shape.

Moreover, since the DLO is standing still at the initial time, suitable initial conditions should be included in the problem, i.e.  $q(u, 0) = q_0$ ,  $\dot{q}(u, 0) = 0$ ,  $u \in [0, L]$ . These conditions can be easily included in the problem by considering that they translate in the same conditions on the control points. i.e.

$$\Lambda_0 q_{k+1} = \Phi_0 \quad (22)$$

$$\Lambda_0 = \begin{bmatrix} B_t(0) \\ B'_t(0) \end{bmatrix}, \quad \Phi_0 = \begin{bmatrix} q_0 \\ 0 \end{bmatrix}$$

Then, a proper set of  $n_s$  time samples  $t_i \in ]0, t_f]$ ,  $i \in 1, \dots, n_s$ , usually called collocation points, are selected, where  $t_f$  is the final time. These time samples are used to generate a set of regression matrices  $\Lambda_i$  and residual vectors  $\Phi_i$  for the corresponding value of  $t_i$ . Those matrices and vectors, together with the initial conditions reported in eq. 22, are then packed as in the following

$$\Lambda_k = \begin{bmatrix} \Lambda_0 \\ \Lambda_1 \\ \vdots \\ \Lambda_{n_s} \end{bmatrix}, \quad \Phi_k = \begin{bmatrix} \Phi_0 \\ \Phi_1 \\ \vdots \\ \Phi_{n_s} \end{bmatrix}$$

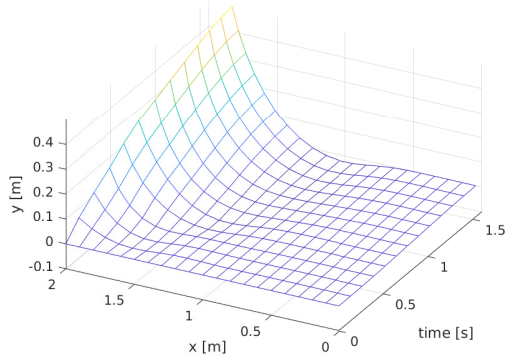
It is clear that the the number  $n_s$  of time samples should be large enough to obtain a suitable set of equations to solve the system, i.e.  $n_s \geq n_u \times n_t$ .

The problem to be solved is now

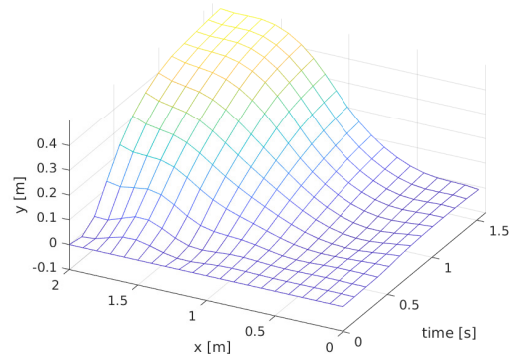
$$\Lambda_k q_{k+1} = \Phi_k \quad (23)$$

that is linear in the unknown  $q_{k+1}$  and provided with a suitable number of equations to admit the trivial solution

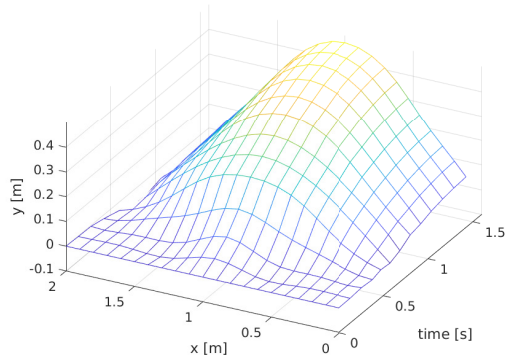
$$q_{k+1} = \Lambda_k^+ \Phi_k \quad (24)$$



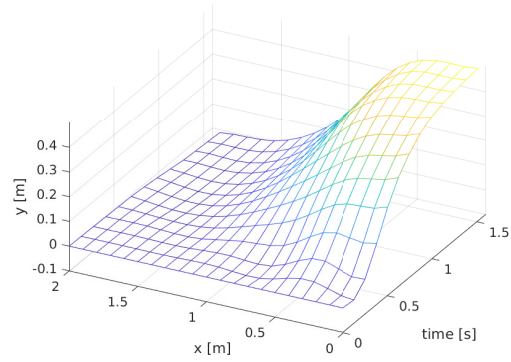
(a) DLO grasped at the distal end.



(b) DLO grasped at 0.5 m from the distal end.



(c) DLO grasped at the middle point.



(d) DLO grasped at 0.5 m from the proximal end.

Fig. 1. Spline interpolation of a some sample manipulation primitives.

where the suffix  $^+$  denotes the matrix pseudoinverse. This solution will provide the value of  $q_{k+1}$  that minimizes the model error in the least square sense, i.e.

$$\min_{q_{k+1}} \|\Lambda_k q_{k+1} - \Phi_k\|$$

The remaining step is now to check how much the new control points  $q_{k+1}$  provided by eq. (24) differ from the actual ones, i.e.  $q_k$ . To do so, a suitable small-enough threshold  $\delta$  is selected and in case the norm of the control points update is lower than this threshold, i.e.

$$\|q_{k+1} - q_k\| < \delta \quad (25)$$

the algorithm is stopped and the last solution  $q_{k+1}$  is taken. Otherwise, the control points are updated, i.e.  $q_{k+1} \rightarrow q_k$ , the new values of  $\Lambda_k$  and  $\Phi_k$  are computed, the new control points  $q_{k+1}$  are provided through eq. (24) and the algorithm is reiterated until the condition (25) is satisfied.

Even though the solution described above is based on the assumption of knowing the external forces  $F$  (the input) and computing the DLO motion described by the control points  $q_{k+1}$  (the output) in the interval  $[t_0, t_f]$ , it is clear that since the system (23) is linear in both  $F$  and  $q_{k+1}$ , it is always possible to exchange, also partially, the input components with the output ones, i.e. it is possible to fix the trajectory of the control points, or some of them, and

compute the corresponding external force needed to move the control points as desired through eq. (23), or to compute the external force needed to move the control points for which the trajectory is fixed and the motion of the remaining control points according to the external forces acting on them by mixing eqs. (23) and (24).

It is worth to notice that the proposed solution may suffer of convergence issues. However, it is possible to state that this issue can be properly solved by suitable selection of the knots sequence adopted to define the spline basis, the number of control points and the number and location of the collocation points, see for example [15] for additional details on these points.

#### IV. SIMULATIONS

In order to evaluate the mathematical framework described in the previous sections, numerical simulations have been implemented in Matlab. The number of basis functions and coefficients for the interpolation along the two dimensions, i.e. the cable length and time, are selected as  $n_u = 9$  and  $n_t = 13$  respectively. These values are selected just to have sufficiently low numbers but a good interpolation capability.

In Fig. 1 the interpolated solutions obtained by means of the proposed spline-based modeling of four manipulation actions where the DLO is grasped on 4 different grasping points are reported. In this simulation, the DLO is assumed

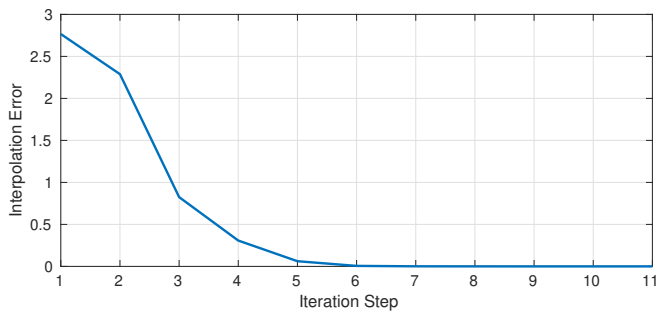


Fig. 2.

to lay on a horizontal  $x$ - $y$  plane in straight position along the  $x$  axis, and the robot moves the grasping point by 0.5 m along the  $y$  direction in 1.6 s. The plots reported in Fig. 1 shows how the manipulation primitive can be seen as a surface parameterized in terms of DLO point along its length and time.

In the iterative computation of the nonlinear system interpolation reported in Sec. III-B, a threshold  $\delta = 1e - 5$  has been used. The plot reported in Fig. 2 reports the convergence rate of the iterative algorithm for a typical manipulation primitive. Despite it is not possible to say that this is the worst case, it represents one the most critical condition among the evaluated ones. It is interesting to note that the error rapidly converges since at the sixth iteration it is already close zero. However, eleven iterations are needed to achieve the required accuracy.

The proposed model execution time has been compared with the simulation of a DLO motion primitive implemented in Simulink using the same spline based model described in Sec. II. While the proposed model takes an average time of 2.6 s to be solved, the simulation implemented in Simulink takes an average time of 52.46 s for the simulation of a single manipulation primitive like the ones reported in Fig. 1, evaluating a mean of 826 steps to solve the system.

## V. CONCLUSIONS

In this paper, the effectiveness of a model-based numerical algorithm to predict the DLO shape manipulated by means of a single arm robot is evaluated through simulation. Despite this is a preliminary work, the proposed algorithm can be exploited in an industrial manufacturing scenario to optimize the robot actions in order to achieve a proper configuration of DLOs. This will enable the robotic manipulator to deal with this kind of objects and will allow the reduction of the production cycle.

Future works will be devoted to the experimental validation of the proposed algorithm, to the exploitation of

this algorithm to optimize the manipulation task, to the extension to two-arms manipulation systems, to integration with an online DLO parameter estimation algorithm and to the extensions of the method to multi-branch DLOs, such as pre-assembled wiring harness used in the automotive and aerospace industries.

## REFERENCES

- [1] R. J. Moreno Masey, J. O. Gray, T. J. Dodd, and D. G. Caldwell, "Guidelines for the design of low-cost robots for the food industry," *Industrial Robot: An International Journal*, vol. 37, no. 6, pp. 509–517, 2010.
- [2] A. Ramisa, G. Alenya, F. Moreno-Noguer, and C. Torras, "Using depth and appearance features for informed robot grasping of highly wrinkled clothes," in *Proc. IEEE Int. Conf. on Robotics and Automation*, 2012, pp. 1703–1708.
- [3] J. Jayender, R. V. Patel, and S. Nikumb, "Robot-assisted active catheter insertion: algorithms and experiments," *The Int. J. of Robotics Research*, vol. 28, no. 9, pp. 1101–1117, 2009.
- [4] J. Pile, G. B. Wanna, and N. Simaan, "Force-based flexible path plans for robotic electrode insertion," in *Proc. IEEE Int. Conf. on Robotics and Automation*, 2014, pp. 297–303.
- [5] A. Shah, L. Blumberg, and J. Shah, "Planning for manipulation of interlinked deformable linear objects with applications to aircraft assembly," *IEEE Tran. on Automation Science and Engineering*, vol. 15, no. 4, pp. 1823–1838, 2018.
- [6] X. Jiang, K.-m. Koo, K. Kikuchi, A. Konno, and M. Uchiyama, "Robotized assembly of a wire harness in a car production line," *Advanced Robotics*, vol. 25, no. 3-4, pp. 473–489, 2011.
- [7] T. Hermansson, R. Bohlin, J. S. Carlson, and R. Söderberg, "Automatic assembly path planning for wiring harness installations," *Journal of manufacturing systems*, vol. 32, no. 3, pp. 417–422, 2013.
- [8] J. Sanchez, J.-A. Corrales, B.-C. Bouzgarrou, and Y. Mezouar, "Robotic manipulation and sensing of deformable objects in domestic and industrial applications: a survey," *The Int. J. of Robotics Research*, vol. 37, no. 7, pp. 688–716, 2018.
- [9] A. Theetten, L. Grisoni, C. Andriot, and B. Barsky, "Geometrically exact dynamic splines," *Computer-Aided Design*, vol. 40, no. 1, pp. 35 – 48, 2008.
- [10] A. Theetten, L. Grisoni, C. Duriez, and X. Merlhiot, "Quasi-dynamic splines," in *Proc. ACM Symposium on Solid and Physical Modeling*, 2007, pp. 409–414.
- [11] R. Zanella, D. De Gregorio, S. Pirozzi, and G. Palli, "Dlo-in-hole for assembly tasks with tactile feedback and lstm networks," in *Proc. Int. Conf. on Control, Decision and Information Technologies (CoDIT)*, 2019, pp. 285–290.
- [12] D. De Gregorio, R. Zanella, G. Palli, S. Pirozzi, and C. Melchiorri, "Integration of robotic vision and tactile sensing for wire-terminal insertion tasks," *IEEE Tran. on Automation Science and Engineering*, vol. 16, no. 2, pp. 585–598, 2018.
- [13] A. Nair, D. Chen, P. Agrawal, P. Isola, P. Abbeel, J. Malik, and S. Levine, "Combining self-supervised learning and imitation for vision-based rope manipulation," in *Proc. IEEE Int. Conf. on Robotics and Automation*, 2017, pp. 2146–2153.
- [14] D. De Gregorio, G. Palli, and L. Di Stefano, "Lets take a walk on superpixels graphs: Deformable linear objects segmentation and model estimation," in *Asian Conference on Computer Vision*, 2018, pp. 662–677.
- [15] C. De Boor, *A practical guide to splines; rev. ed.*, ser. Applied mathematical sciences. Berlin: Springer, 2001.
- [16] O. Nocent and Y. Remion, "Continuous deformation energy for dynamic material splines subject to finite displacements," in *Computer Animation and Simulation 2001*, N. Magnenat-Thalmann and D. Thalmann, Eds. Vienna: Springer Vienna, 2001, pp. 87–97.