Symplectic Integration for Multivariate Dynamic Spline-Based Model of Deformable Linear Objects

Published Version:
Symplectic Integration for Multivariate Dynamic Spline-Based Model of Deformable Linear Objects / Khalifa A.; Palli G.. - In: JOURNAL OF COMPUTATIONAL AND NONLINEAR DYNAMICS. - ISSN 1555-1415. - ELETTRONICO. - 17:1(2022), pp. 011001.1-011001.9. [10.1115/1.4052571]

Availability:
This version is available at: https://hdl.handle.net/11585/841310 since: 2021-12-10

Published:
DOI: http://doi.org/10.1115/1.4052571

Symplectic Integration for Multivariate Dynamic Spline-Based Model of Deformable Linear Objects

Alaa Khalifa
Department of Electrical, Electronic and Information Engineering,
University of Bologna, Italy
On leave from Department of Industrial Electronics and Control Engineering,
Faculty of Electronic Engineering, Menoufia University, Egypt
Email: alaa.khalifa2@unibo.it

Gianluca Palli
Department of Electrical, Electronic and Information Engineering,
University of Bologna, Italy
Email: gianluca.palli@unibo.it

Deformable Linear Objects (DLOs) such as ropes, cables, and surgical sutures have a wide variety of uses in automotive engineering, surgery, and electromechanical industries. Therefore, modeling of DLOs as well as a computationally efficient way to predict the DLO behavior are of great importance, in particular to enable robotic manipulation of DLOs. The main motivation of this work is to enable efficient prediction of the DLO behavior during robotic manipulation. In this paper, the DLO is modeled by a multivariate dynamic spline, while a symplectic integration method is used to solve the model iteratively by interpolating the DLO shape during the manipulation process. Comparisons between the symplectic, Runge-Kutta and Zhai integrators are reported. The presented results show the capabilities of the symplectic integrator to overcome other integration methods in predicting the DLO behavior. Moreover, the results obtained with different sets of model parameters integrated by means of the symplectic method are reported to show how they influence the DLO behavior estimation.

1 Introduction

Many of the objects we handle every day are highly deformable and with prevalent plastic behavior. However, humans manipulate those objects naturally, with high dexterity, and without any particular issue. In facts, manipulating these deformable objects has a wide variety of uses in domestic facilities and healthcare, such as robotic surgery [1], assistive dressing, garment sorting or folding clothing [2], [3]. Moreover, it is involved in manufacturing, aerospace [4], automotive, and electromechanical industries generally [5], [6].

On the contrary, the manipulation of deformable objects is still a challenging activity for robots. This is the main reason why many assembly procedures involving such deformable objects are still performed manually. One of the main reasons why robots have such limitations in deformable object manipulation is due to their complex behaviors, unpredictable initial configuration, and limited capability in measuring their state.

A thoughtful survey can be found in [7] that focuses on deformable object manipulation by robots in industrial and domestic applications. Actually, the dynamics of deformable objects is complex and nonlinear. Therefore, the state estimation of a deformable object is challenging, and the forward prediction is expensive. Robust and effective methods to manipulate deformable objects and predict their behavior remain extremely difficult to build, despite the several applications and attempts made by the robotics community [7].

In this work, the numerical integration of Deformable Linear Objects (DLOs), such as strings, cables, electric wires, catheters, ropes, and so on, is addressed, since efficient solutions to this problem will enable the implementation of robotized solution in many relevant subfields of the large and diverse industrial manufacturing. DLO manipulation is fundamental in automotive manufacturing, e.g. for wiring harness preparation and electrical cable installation inside the vehicle structure, as well as in medical surgery, e.g. in suturing in which a flexible wire in a straight configuration needs to go to a knot [8], and have a vital role in other fields such as architecture and power distribution. The relevant work on DLO manipulation can be split into three overlapping categories: modeling, simulation, and planning. Various manipulation tasks such as knot tying [9], rope untangling [10], string insertion [11], and shape manipulation [12] can be executed on DLOs.

A model of DLO deformation/flexibility is needed in order to predict its behavior successfully. This model implies that both the geometry and the mechanical behavior of the concerned parts can be represented accurately. Once the model is properly defined, a computationally efficient way is needed to evaluate it over time or to solve queries of motion planning. Virtual prototyping is used to reduce development costs and to boost consistency. Also, it enables the early detection of possible problems. It allows for studying the efficiency of assembly. The most distinctive characteristic of DLOs is the variations in shape that they undergo following the influence of forces and environmental constraints. Such shape variations may be the purpose of the planning itself, for example if they are caused by obstacles along the path [13]. Overall, DLO modeling is a major and complex challenge, with a broad range of applications. The PDE model can be established [14, 15, 16, 17, 18], and the associated structure-preserving method can be developed for the dynamic problems of the DLOs. Various research in the literature focused on modeling and manipulating DLOs for various purposes and many models and strategies were created. There are several various approaches for physically modeling DLOs such as Mass-spring [19], Multi-body [20], Elastic rod [21], Dynamic spline [22], Finite element [23], and other models. The methodology, advantages and disadvantages of each model method are discussed in detail in [24]. The dynamic spline is one of these methods which provide a good theoretical basis, continuous model, higher authenticity. A geometrically reliable model of DLOs is generated and adopted in [25] to execute numerical simulations on the motion of the object under gravity and during environmental interaction. For interaction simulation, a linearized spline-based DLO model named quasi dynamic splines is created in [26].

Owing to the inherent trade-off between precision and real-time capability, it is hard to deal with DLOs due to their complex model, resulting then in time-consuming integration processes to predict their behavior. To solve numerically the differential equations, many types of integrators can be found in literature, such as Runge-Kutta, Euler, Taylor type integrators and many others. Despite their general applicability, the main problems associated with the use of these integrators are that they are often inefficient, and prone to instability. In [27] a model of DLO is implemented using a multivariate dynamic spline. The integration process is achieved using the traditional type Runge-Kutta method. This integration process is very time consuming and provide unstable results with many stuck situations. Also, it is not practical for a long-time experiment. Therefore, it is important to find another integration method to reduce simulation time.

The dynamics of a mechanical system, such as a DLO, can be easily represented by means of Hamiltonian equations. It is also known that a symplectic transformation is the solution of a Hamiltonian system [28]. The numerical solutions obtained by several numerical techniques, such as the Runge-Kutta method and the primitive Euler scheme, are not energy-preserving map, resulting in spuriously damped or excited behaviors.

On the other hand, the symplectic integrator proposed in [29] exploits the energy-preserving symplectic transformation to avoid spuriously damped or excited solutions. The symplectic integrator scheme has been widely applied to the calculations of the long-term evolution of chaotic Hamiltonian systems [30], and [31]. In the actions of a Hamiltonian system, the symplectic integrator does not generate a secular truncation error. The number of force function evaluations of the fourth-order symplectic integrator is smaller than the ones of the Runge-Kutta integrator of the same order. The energy conservation and the long-time stability for the symplectic scheme are investigated and verified in [29], [32], [16], and [33]. There are many merits for symplectic integrator which are discussed and outlined in deeper detail in [34], [35], and [36]. For these advantages, the use of the symplectic integrator is investigated in this paper for DLO simulation instead of the conventional integrators. To this end, a comparison between the results obtained from this symplectic integrator and other types of integrators will be executed.

The main objective of this paper is to compare the properties and results obtained by the symplectic integrator with the ones obtained with other numerical methods. This activity is justified by the fact that, at the best of our knowledge, no data are available in literature about the performances of numerical integration algorithms for DLOs. For the DLO modeling, multivariate dynamic splines are used in this paper. A comparison between the symplectic, Runge-Kutta, and Zhai methods is performed in two cases. In the first case, the effect of gravitational and internal forces only are considered, while in the second case the application of an external force in addition to the previously mentioned forces is considered.

The remainder of this article is arranged as follows. Sec. 2 outlines the key features of the mathematical model of the DLO. The conversion from Lagrangian into Hamiltonian and the symplectic integrator are discussed in Sec 3. Moreover, Sec. 4 introduces the preliminary simulation results and comparisons. Finally, conclusions and future work are draft out in Sec. 5.

2 Dynamic Model of the DLO

A third-order spline basis can effectively represent the shape of the DLO. It is a function of a free coordinate u. This coordinate u represents the position on the DLO which is equal to zero at the first endpoint. Also, u equals L at the opposite endpoint, provided that L is the length of the DLO. This can be written mathematically as:

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

where $q(u) = (x(u), y(u), z(u), \theta(u)) = (r(u), \theta(u))$ is the fourth-dimensional configuration functional space of the DLO. It includes the three linear coordinates x, y, z of the

DLO position at point u, and the DLO's axial twisting θ. Also, $b_i(u)$ is the i^{th} spline polynomial basis employed to describe the shape of the DLO. Moreover, q_i are properly chosen coefficients, typically called control points, used to correctly interpolate the shape of the DLO through the $b_i(u)$ basis functions. n_u is the number of control points.

For a variety of reasons, this DLO mathematical model is highly successful. Firstly, the spatial derivatives calculation is straightforward, i.e.

$$q^{(j)}(u) = \sum_{i=1}^{n_u} b_i^{(j)}(u) q_i, \tag{2}$$

where $q^{(j)}$ is the j^{th} derivative of q. In fact, it can be defined by the same coefficients and easy-to-compute derivatives of $b_i(u)$. Second, the spline basis proprieties guarantee that the DLO model curvature, which represents the DLO's physical behavior [37], is minimized. Third, this model enables the representation of a generalized nonlinear function with smoothness characteristics as a linear combination of the nonlinear function basis $b_i(u)$, which relies on the free variable u only, by the linear coefficients q_i.

Referring to the system's Lagrange equations, the DLO's dynamic model can be expressed as a function of the control points q_i as:

$$\frac{d}{dt}\left(\frac{\partial K}{\partial \dot{q}_i}\right) = F_i - \frac{\partial U}{\partial q_i}, \qquad i = 1, 2, \ldots, n_u, \tag{3}$$

where K is the total kinetic energy, F_i is the external force that is applied to the i^{th} control point, and U is the total potential energy acting on the DLO. This potential energy is generated due to the influence of gravity, stretching, torsion, and bending effects on the DLO.

2.1 Kinetic Energy of the DLO

Owing to the translational motion of the control points and the rotational motion of the cross sections, the total kinetic energy is generated. In order to express this total kinetic energy, a function of the control points q_i can be used as :

$$K = \frac{1}{2}\int_0^L \frac{dq^T}{dt} J \frac{dq}{dt}\, ds, \quad J = \begin{bmatrix} \mu & 0 & 0 & 0 \\ 0 & \mu & 0 & 0 \\ 0 & 0 & \mu & 0 \\ 0 & 0 & 0 & I \end{bmatrix}, \tag{4}$$

where J is the DLO's generalized density matrix, ds is the displacement element and can be computed from $ds = \|r'(u)\|\, du$, μ denotes the linear density, and I denotes the polar moment of inertia.

According to the procedure explained in [38], we can drive:

$$\frac{d}{dt}\left(\frac{\partial K}{\partial \dot{q}_i}\right) = \sum_{j=1}^{n_u} M_{ij} \ddot{q}_j. \tag{5}$$

Then, by expanding this definition to the overall system, the total inertial forces of the DLO can be written as $M\ddot{q}$, in which M denotes the inertia matrix of the DLO and $\ddot{q}$ is a vector that represents the control point accelerations.

2.2 Potential Energy of the DLO

The total potential energy U is made up of the gravitational effect and the strain energy of the DLO due to stretching, torsion and bending. Although the derived energy of gravity is quite simple, the concept of strain energy plays a crucial role in the modeling of DLO. By introducing the strain vector $\varepsilon = [\varepsilon_s, \varepsilon_t, \varepsilon_b]$ that includes ε_s to represent the stretching term, ε_t to denotes the torsional term, and ε_b to denotes the bending term such that:

$$\begin{aligned} &\varepsilon_s = 1 - \|r'\|, \quad \varepsilon_t = \theta' - \gamma, \quad \varepsilon_b = \frac{\|C\|}{\|r'\|^3}, \\ &C = r' \times r'', \quad \gamma = \frac{C^T r'''}{\|C\|^2}, \end{aligned} \tag{6}$$

where r and θ are the strain's linear and the torsional component, respectively. The strain energy can therefore be written as:

$$U = \frac{1}{2}\int_0^L (\varepsilon - \varepsilon_0)^T H (\varepsilon - \varepsilon_0)\, ds = \frac{1}{2}\int_0^L \varepsilon_e^T H \varepsilon_e\, ds, \tag{7}$$

where

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

ε_0 is the DLO's plastic strain, which enables to consider the plasticity of the material. ε_e is called the residual strain that equals $\varepsilon - \varepsilon_0$. Also, H is the element stiffness matrix, D is the cross-section diameter of the DLO. Moreover, E and G denote the Young's modulus and the shear modulus of the DLO's material, respectively.

It is possible to write the right-side term of Eq. (3) as:

$$P_i = -\frac{\partial U}{\partial q_i} = -\frac{1}{2}\int_0^L \frac{\partial \varepsilon_e^T H \varepsilon_e}{\partial q_i} ds = -\int_0^L \frac{\partial \varepsilon^T}{\partial q_i} H \varepsilon_e\, ds, \tag{8}$$

where P_i represents the elastic forces because of the DLO deflection.

2.3 Dynamic Model of the DLO

To write the overall dynamic model of the DLO, the Eqs. (3), (5), and (8) are extended to the whole control points

$$M\ddot{q} = F + P \tag{9}$$

where F refers to the vector that contains all external forces, including gravity, while P represents the vector that contains all elastic forces.

At every simulation step, the system can be solved by utilizing a simple LU decomposition. Moreover, at each time step, the accelerations are integrated to get the control points velocities and positions. Several integration methods can be used, but some problems may arise, including numerical instability, large time required for the integration process and the unsuitability for long-time predictions. These problems can be mitigated by using the symplectic integrator, as will be shown in the following. The next section will discuss this symplectic integrator and its requirements in more details.

3 Symplectic Integrator

To solve numerically the differential equations, we decided to use the symplectic integrator because of its advantages with respect to other methods as will be shown in the following. These subsections will discuss in detail calculating the Hamiltonian from Lagrangian and the construction of the symplectic integrator.

3.1 Conversion from a Lagrangian to a Hamiltonian

Symplectic integrator is widely used in nonlinear dynamics. It is designed as the numerical solution for Hamilton's equations, which provided as:

$$\dot{p} = -\frac{\partial H_m}{\partial q} \quad \text{and} \quad \dot{q} = \frac{\partial H_m}{\partial p}, \tag{10}$$

where p denotes the momentum coordinates, q refers to the position coordinates, and H_m is the Hamiltonian which can be found from:

$$H_m(p,q) = K(p) + U(q), \tag{11}$$

where K and U denote the kinetic and potential energy, respectively. The set of position and momentum coordinates (q,p) are named canonical coordinates. In our case, the first step to use the symplectic integrator is to make a conversion from Lagrangian into Hamiltonian [39]. This conversion is achieved according to the following procedure.

Provided a Lagrangian L_g as a function of the generalized coordinates q_i and generalized velocities $\dot{q}_i$, where $L_g(q_i,\dot{q}_i) = K(\dot{q}_i) - U(q_i)$, the Hamiltonian can be calculated according to the following steps:

1. By differentiating the Lagrangian L_g with respect to the generalized velocities $\dot{q}_i$, the momenta p_i are determined:

$$p_i(q_i,\dot{q}_i) = \frac{\partial L_g}{\partial \dot{q}_i} = \frac{\partial \mathrm{K}}{\partial \dot{q}_i}. \tag{12}$$

2. By inverting the expressions in the former step, the velocities $\dot{q}_i$ are formulated in terms of the momenta p_i.

$$p_i = J\dot{q_i} \qquad \text{so, } \dot{q}_i = J^{-1}p_i, \tag{13}$$

where $J^{-1} = \begin{bmatrix} 1/\mu & 0 & 0 & 0 \\ 0 & 1/\mu & 0 & 0 \\ 0 & 0 & 1/\mu & 0 \\ 0 & 0 & 0 & 1/I \end{bmatrix}$.

3. Using the Lagrangian relation ($L_g = K - U$), we can conclude that:

$$L_g = \frac{1}{2}\int_0^L \dot{q}^T J\dot{q}\, ds - U. \tag{14}$$

The velocities $\dot{q}_i$ are then substituted from Equ. (13),

$$L_g = \frac{1}{2}\int_0^L p^T J^{-1} p\, ds - U. \tag{15}$$

4. The Hamiltonian is determined by employing the typical definition of H_m as the Legendre transformation of L_g:

$$H_m = \sum \dot{q}_i \frac{\partial L_g}{\partial \dot{q}_i} - L_g = \sum \dot{q}_i p_i - L_g. \tag{16}$$

Substitution for $\dot{q}_i$ from Equ. (13) and the Lagrangian L_g from Equ. (15) into Equ. (16), will lead to:

$$H_m = \frac{1}{2}\int_0^L p^T J^{-1} p\, ds + U. \tag{17}$$

The last equation is the require one as it is equivalent to the Hamiltonian equation that stated in Equ. (11).

3.2 Symplectic Integrator of the fourth-order

Forest [34] and Neri [36] introduced the generalized form of the fourth-order symplectic integrator as:

$$
\begin{aligned}
q_1 &= q_0 + c_1\tau\tfrac{\partial K}{\partial p}(p_0), & p_1 &= p_0 - d_1\tau\tfrac{\partial U}{\partial q}(q_1),\\
q_2 &= q_1 + c_2\tau\tfrac{\partial K}{\partial p}(p_1), & p_2 &= p_1 - d_2\tau\tfrac{\partial U}{\partial q}(q_2),\\
q_3 &= q_2 + c_3\tau\tfrac{\partial K}{\partial p}(p_2), & p_3 &= p_2 - d_3\tau\tfrac{\partial U}{\partial q}(q_3),\\
q_4 &= q_3 + c_4\tau\tfrac{\partial K}{\partial p}(p_3), & p_4 &= p_3 - d_4\tau\tfrac{\partial U}{\partial q}(q_4),
\end{aligned}
\tag{18}
$$

where τ is the time step-size, q_0 and p_0 are the initial values, and q_4 and p_4 are the numerical solution after τ. Moreover, c_i and d_i are numerical coefficients which can be determined uniquely from:

$$
\begin{gathered}
c_1 = c_4 = \tfrac{1}{2(2-\beta)}, \quad c_2 = c_3 = \tfrac{(1-\beta)}{2(2-\beta)},\\
d_1 = d_3 = \tfrac{1}{(2-\beta)}, \quad d_2 = \tfrac{-\beta}{(2-\beta)}, \quad d_4 = 0,\\
\text{where } \beta = 2^{1/3}.
\end{gathered}
\tag{19}
$$

The reader can refer to the reference [35] for the derivation and the values of the numerical coefficients.

The number of force function evaluations $\frac{\partial U}{\partial q}$, that is the most time-expensive operation in the integration process, is three rather than four since $d_4 = 0$. On the other hand, four force function evaluations are required in the conventional fourth-order Runge-Kutta integrator. Thus, the CPU time can be decreased by approximately 25 percent with the symplectic integrator with respect to the fourth-order Runge-Kutta method.

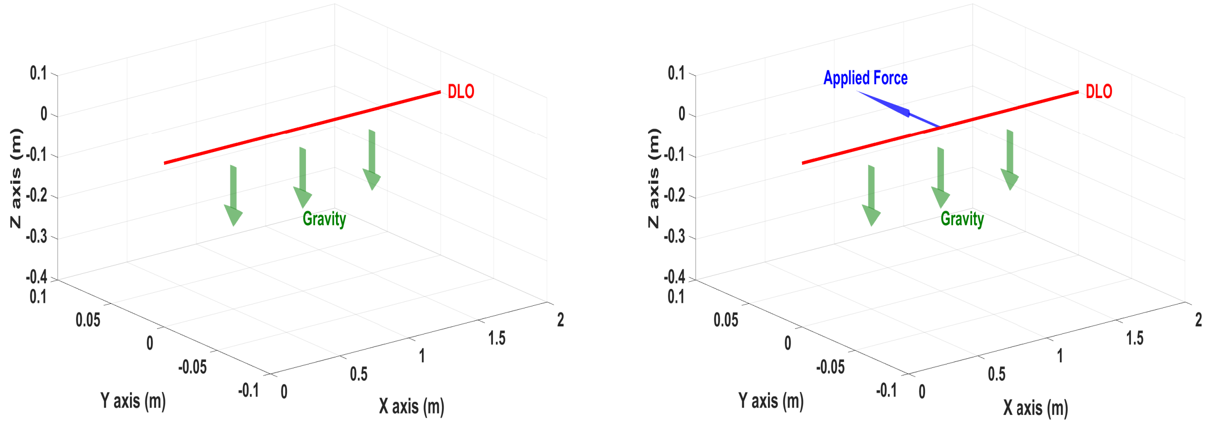
Other than the energy-preserving property, a remarkable characteristic of the symplectic integrator is that the aggregation of the truncation error in the total energy has not a secular component and the positional errors rise in proportional to the first order of time [28]. On the other hand, conventional integrators generate a secular energy error. Moreover, the error due to position discretization increase with the square of time. The symplectic integrator is thus well suited to the long-time study of a dynamic system. Generally, in the actions of a Hamiltonian system, the symplectic integrator does not generate a secular truncation error [28].

4 Simulation Results

Numerical simulations have been carried out in MATLAB for the assessment of the mathematical framework discussed in the preceding sections. The issue that is aimed in this section is determining the DLO state evolution, i.e. the control points evolution over time, while influenced by a internal elastic and inertial forces as well as external forces like gravitational force or contacts forces generated by obstacles. In the case of contacts, we can reshape the system as a constrained dynamic system in which the Lagrange approach for constraints is adopted. Just add some constraints to the solution of the dynamic system. This will not change the problem, but just increase the number of dynamic equations that existed in the model. In [25], the solution of dynamic equations representing the DLO dynamics including contacts is reported. The same approach can be applied also to the symplectic integrator by changing the dynamic equations accordingly. The simulations have been performed using the dynamic model in Eq. (9). It is worth mentioning that all simulations of this work have been performed on Ubuntu 18.04.5 LTS operating system, processor intel core i5-3210M CPU@2.50GHz x 4, RAM 8GiB.

In our simulations, DLO's material is assumed to be aluminum, and its length L is 2 meters. The DLO has a circular cross-section with a diameter D that equals 2 millimeters. The number of control points is selected as $n_u = 9$. Also, the number of sample points along the DLO is chosen to be n_s=101. These values are chosen to be sufficiently low but provide a good interpolation capability. During these simulations, the DLO is supposed to be placed initially on a straight position along the x-axis, and the two extreme endpoints are constrained to move along the x-axis and attached by springs with stiffness $(K_x = 10\ kN/m)$ to their initial position. In other words, one endpoint of the DLO is located at $(0,0,0)$, while the opposite endpoint is located at $(L,0,0)$. Two simulation scenarios will be used in this section. In the first scenario, the DLO is affected by the internal forces and the gravitational force only as shown in Fig. 1a. While in the second scenario, the DLO is affected by the previously mentioned forces in addition to an external sinusoidal force which is applied to the DLO center in the Y-direction as illustrated in Fig. 1b.

Fig. 2, and 3 illustrate the computed solutions accomplished utilizing the spline-based model using symplectic integrator where the DLO is affected by the gravity and the internal forces only without any other external forces. Fig. 2 presents the x-component of each control point, while Fig. 3 shows the z-component of each control point. This simulation is achieved for 10 seconds using 2 milliseconds as a time step-size τ. This time step-size is the largest value that preserves the stability of the integration method. It is worth mentioning that, during the whole period, the y component for each control point is equal to zero as the DLO is positioned along the x-axis and the external force is not applied yet. In Fig. 3, due to the symmetry of the DLO, each two control points' trajectories are located above each other in one curve except the middle point. A video that shows this numerical simulation exists in the link [1]. Fig. 4 shows the sequence of trajectories that illustrate the DLO motion starting from the initial position according to the first scenario, i.e. the DLO is affected by the internal forces and the gravitational force only without any other external forces.



(a)

(b)

Fig. 1. Schematic diagram of the DLO in which it is affected by: a) gravitational force, and b) gravitational force and the DLO's center is affected by an external force in the Y direction.

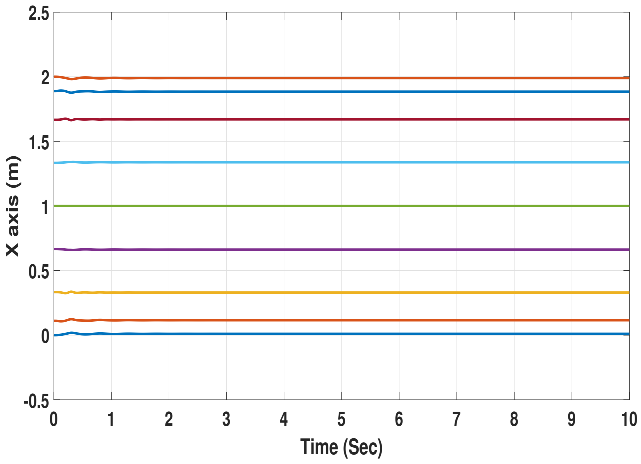


Fig. 2. X component of each control point of the DLO.

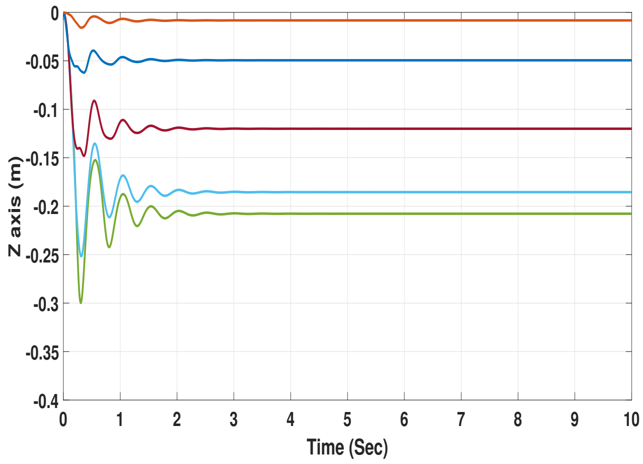


Fig. 3. Z component of each control point of the DLO.

The simulation is performed again in the case of applying an external force on a DLO point. We assumed that the external force takes the shape of a sinusoidal trajectory. This external force is applied to the center of the DLO in the Y-axis direction. Fig. 5, 6, and 7 show the x, y, and z components, respectively. The solution discussed here assumes the knowledge of the external forces F (the input). The computation will give the control points q (the output) that describe the DLO motion during the simulation interval. A video that illustrates this numerical simulation, in case of applying an external force, is found in the link [2]. These results prove the ability of the spline-based dynamic model and the symplectic integrator to represent the state evolution of the DLO accurately. Fig. 8 shows the sequence of trajectories that illustrate the DLO motion starting from the initial position according to the second scenario, i.e. the DLO is affected by the internal forces, the gravitational force, and an external sinusoidal force applied to the DLO center in the Y-direction.

The DLO model simulation time of the symplectic integrator is compared with the ones of both the classical Runge-Kutta integrator and Zhai integrator. Zhai integrator is a new simple fast explicit time integration method, the reader can refer to [40] for the related details and advantages of Zhai integrator. In MATLAB/Simulink, the three integrators are implemented by using the same spline-based model defined in Sec. 2. In the case the DLO is affected by internal forces and gravity only, Table 1 reports the comparison between the three integrators' simulation time in seconds for a certain period (10 sec) with 2 milliseconds as a time step-size τ. The results show that the symplectic integrator is faster than both the Runge-Kutta and Zhai integration methods. It is worth mentioning that the Zhai method needs optimization for two parameters [40] . However, we made a manual tunning to these parameters. This may be the reason for the notable slowness of Zhai compared to the Runge-Kutta method.

Table 2 presents the comparison between the three integrators' computation time in seconds for a simulation of 10 seconds in the case where the DLO is affected by the exter-

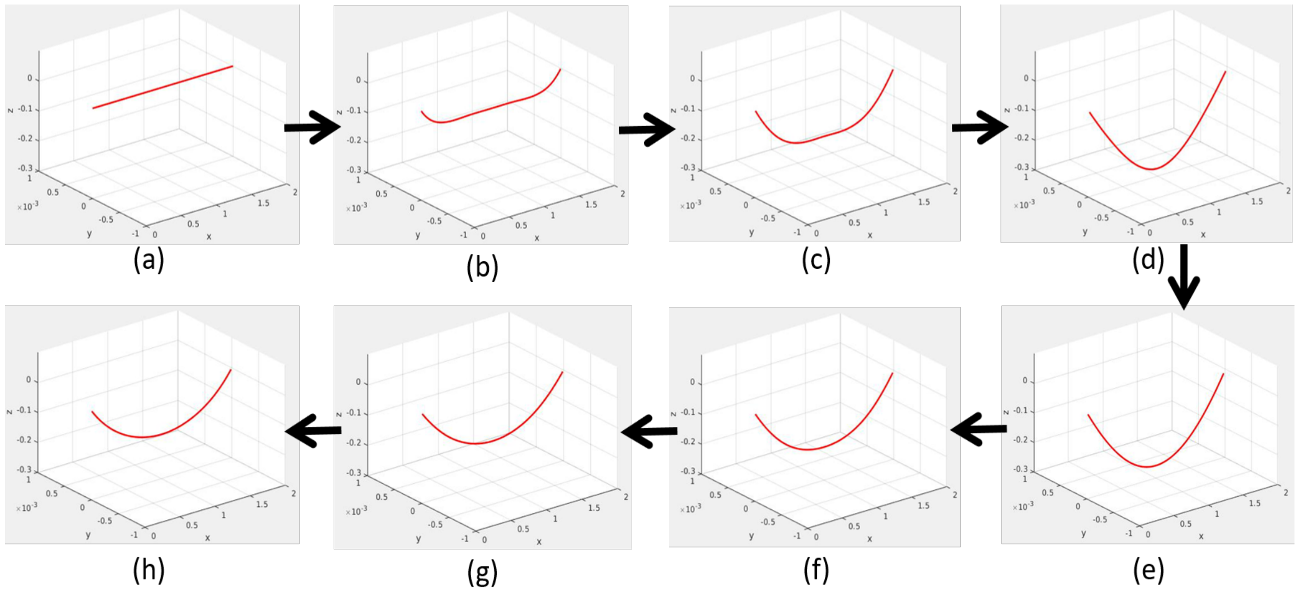


Fig. 4. Sequence of trajectories to show the DLO motion starting from the initial position according to the first scenario.

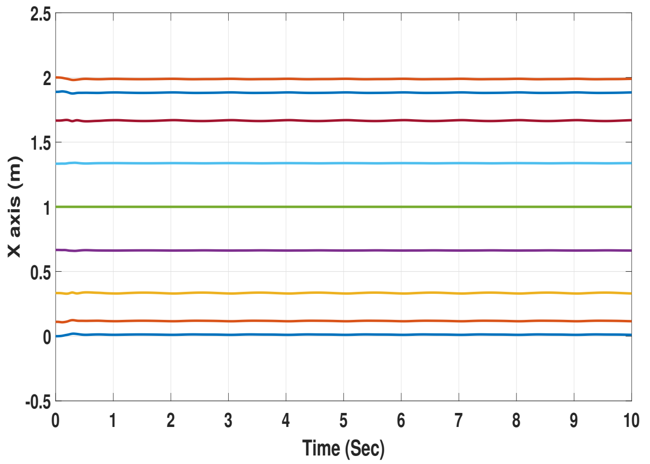


Fig. 5. X component of each control point of the DLO in which its center is affected by external sinusoidal force.

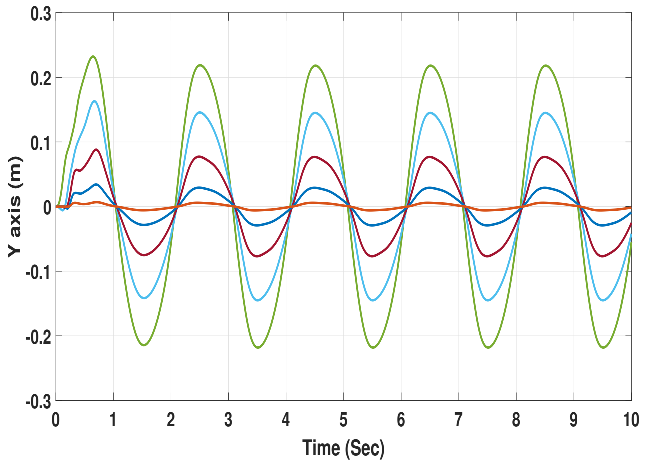


Fig. 6. Y component of each control point of the DLO in which its center is affected by external sinusoidal force.

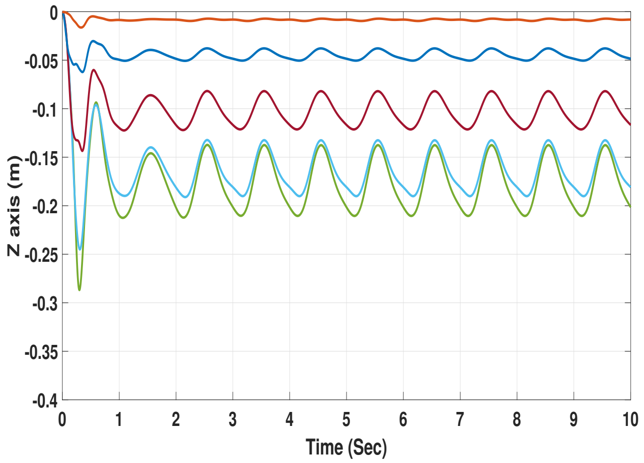


Fig. 7. Z component of each control point of the DLO in which its center is affected by external sinusoidal force.

nal force in addition to the previously mentioned forces. The results show that the symplectic integrator is able to solve the problem while, on the other hand, the Runge-Kutta and Zhai integrators have been stuck and could not compute any solution. It is worth mentioning that, in this work, we did not investigate the reason why other methods are unstable.

The simulation implemented in MATLAB, using the fastest integrator, takes an average time of 470 seconds for the simulation of 10 seconds like the one reported in Fig. 2 and 3. This long execution time occurs due to two reasons. The first reason is the specifications of the processing unit or the PC. The second reason is the implementation on MATLAB which is not really efficient. To avoid this reason, another simulation is repeated using C++ language. In these simulations and in the related comparisons, the Zhai integrator is excluded as it is the slowest and it looks not really efficient in solving the DLO dynamics. Hence, the current com-

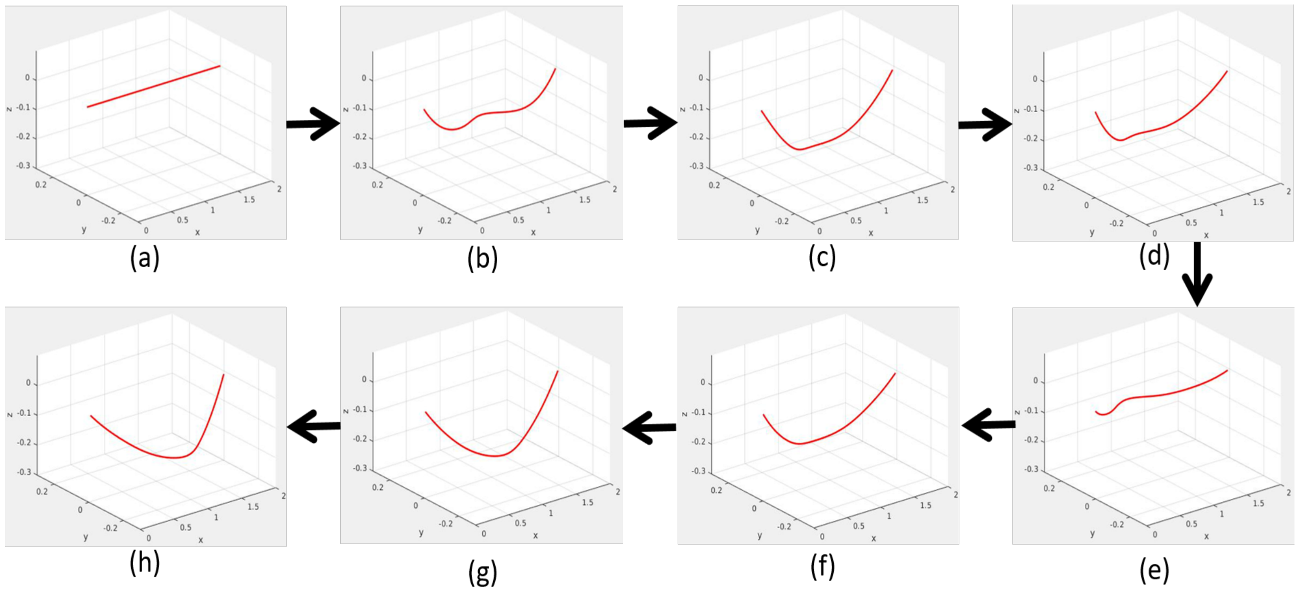


Fig. 8. Sequence of trajectories to show the DLO motion starting from the initial position according to the second scenario.

parison will be focused on the symplectic and the Runge-Kutta integration methods and their implementation in C++. Tables 3 and 4 introduce the comparison between the two integrators' execution time in the case of internal forces only and sinusoidal external force, respectively. From Tables 3 and 4, it can be concluded that the symplectic integrator gives the fastest execution time. Also, the C++ execution time is reduced to be one-third of the execution time of MATLAB results.

To check the effect of changes in the DLO model granularity, n_u and n_s on the execution time, simulations on both MATLAB and C++ are performed. Tables 5 and 6 present MATLAB results of the execution time by changing n_u and n_s for both cases of internal forces only and sinusoidal external force, respectively. Each cell in the two tables introduces the execution time in seconds for a simulation period of 10 seconds. Likewise, Table 7, and 8 illustrate C++ results of execution time with changing n_u and n_s. From these tables, we conclude that C++ is faster about three times more than MATLAB implementations. So, C++ implementation with symplectic integrator will be better for the practical implementation of the simulation system. Moreover, reducing n_u and/or n_s will require less execution time, but it will reduce interpolation precision in representing the DLO. It is worth mentioning that the time step size is chosen to be unique and equal to 0.8 milliseconds for all of these comparisons. As n_u and n_s are increased, more calculations are required and hence the integration time increases. The same step size is selected for all comparisons for fairness.

The model with the highest resolution ($n_s = 256$ and $n_u = 19$) is considered as the reference model. Each model resolution considered in this work is compared to the reference model and the error is computed. For each resolution, the DLO position is measured at 10 fixed points along the DLO. Moreover, the error over time is computed by considering 20 equidistant time intervals. Fig. 9 shows the evolution of this error over the time and the DLO length. It is noticed that the maximum error occurred at the center of the DLO and the error decreases over time. The evolution of the error with different values for n_s and n_u is presented in Fig. 10. As n_u increases, the error decreases and vice versa, while n_s has no noticeable effect on the error. Moreover, as the number of control points n_u increases, the error decreases over time and with respect to the position along the DLO as indicated in Fig. 11 and 12, respectively.

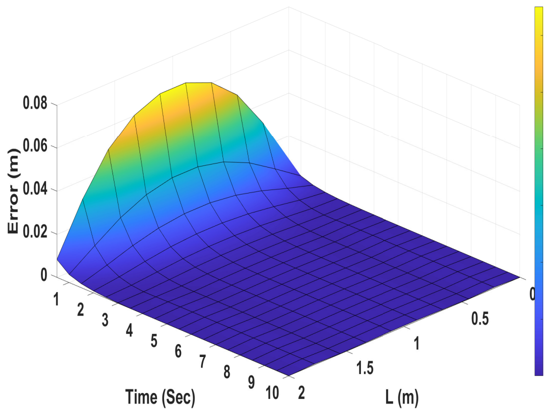


Fig. 9. Evolution of the error over time and DLO length.

5 Conclusions

This paper reports a comparison among integration methods to solve the dynamic model of Deformable Linear Objects. The adopted DLO model is based on the multivariate dynamic spline and employs the symplectic integrator. This symplectic integrator received particular attention

Table 1. MATLAB execution time for solving the DLO model employing different integrators without applying an external force.

Simulation time	Symplectic	Runge-Kutta	Zhai
0.1	6.66	9.43	34.85
0.2	11.39	16.42	67.57
0.3	16.14	22.43	96.69
0.4	20.86	28.41	120.45
0.5	25.60	34.62	145.42
0.6	30.42	41.29	166.22
0.7	35.17	48.62	186.46
0.8	39.92	56.07	206.70
0.9	44.61	62.59	226.81
1	49.30	68.89	247.39
2	97.18	140.00	478.94
3	144.21	210.41	685.46
4	191.20	278.22	904.89
5	238.24	347.53	1127.37
6	285.31	418.16	1348.17
7	332.29	487.72	1542.90
8	379.23	555.20	1733.71
9	426.17	624.36	1970.40
10	473.18	697.04	2204.93

Table 2. MATLAB execution time for solving the DLO model employing different integrators in case of applying an external force.

Simulation time	Symplectic	Runge-Kutta	Zhai
0.1	6.83		
0.2	11.75		
0.3	16.70		
0.4	21.64		
0.5	26.60		
0.6	31.56		
0.7	36.55		
0.8	41.48		
0.9	46.44		
1	51.40	Unstable	Unstable
2	101.14		
3	150.73		
4	200.30		
5	249.89		
6	298.60		
7	346.63		
8	394.63		
9	442.50		
10	490.59		

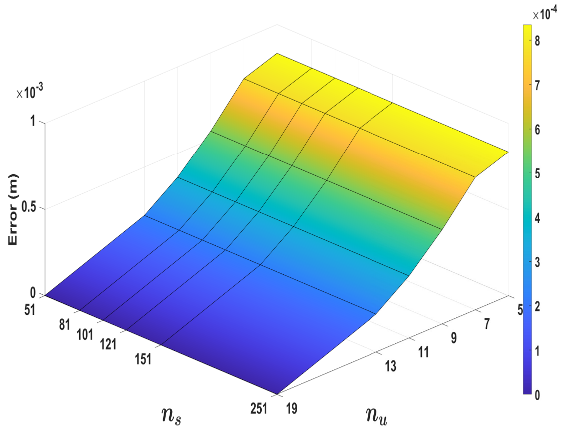


Fig. 10. Evolution of the error over n_s and n_u points.

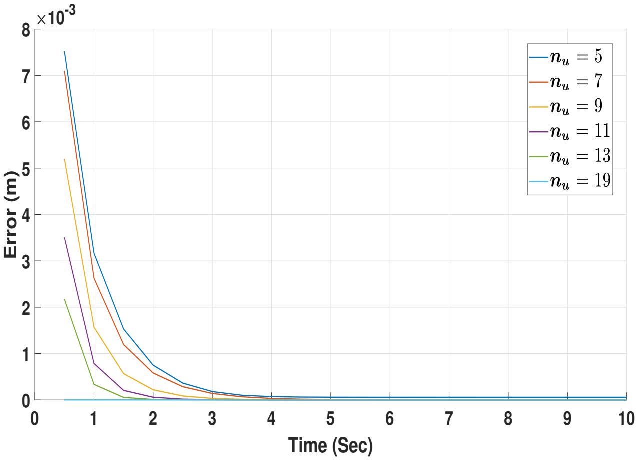


Fig. 11. Evolution of the error over time at different n_u points.

due to its advantages over alternative methods in solving the dynamic equations of Hamiltonian systems. Simulations on MATLAB and using C++ code are performed to compare the performance of the symplectic integrator against the Runge-Kutta and Zhai integrators. The results prove that the symplectic integrator is the fastest and the most stable integrator among the considered ones. Moreover, the proposed comparison shows that C++ is about three times faster than the corresponding MATLAB implementation. Hence, a spline-based model with a symplectic integrator looks to be the best choice for practical implementation. Finally, comparisons are performed to check the effects of changing the model granularity. These results show that reducing the control points and sample points will reduce the execution time, but it will reduce the interpolation precision in representing the DLO behavior.

In future work, practical experiments will be used to validate the spline-based model and the results obtained with

Table 3. C++ execution time for solving the DLO model employing different integrators without applying an external force.

Simulation time	**Symplectic**	**Runge-Kutta**
0.1	2.04	5.88
0.2	3.85	10.44
0.3	5.65	13.90
0.4	7.55	16.75
0.5	9.38	19.93
0.6	11.17	23.09
0.7	12.95	26.43
0.8	14.73	29.82
0.9	16.56	33.00
1	18.35	36.39
2	36.57	69.57
3	54.67	102.85
4	72.88	136.00
5	90.99	169.06
6	109.11	201.95
7	127.30	235.07
8	145.86	268.21
9	164.16	301.24
10	182.46	334.40

Table 4. C++ execution time for solving the DLO model employing different integrators in case of applying an external force.

Simulation time	**Symplectic**	**Runge-Kutta**
0.1	1.98	5.45
0.2	3.98	9.85
0.3	5.92	13.63
0.4	7.93	16.88
0.5	9.99	20.12
0.6	11.91	23.42
0.7	13.97	26.81
0.8	15.99	30.52
0.9	17.90	34.53
1	19.90	38.17
2	40.10	76.95
3	60.16	115.53
4	80.41	154.16
5	101.47	193.16
6	123.51	231.97
7	146.46	270.46
8	167.34	309.58
9	189.53	349.07
10	210.46	388.04

Table 5. Effects of changing the model parameters $(n_u \& n_s)$ on MATLAB execution time for solving the DLO model employing the symplectic integrator without applying an external force.

n_u	n_s					
	51	**81**	**101**	**121**	**151**	**251**
5	488.28	688.22	825.99	960.52	1288.93	2094.12
7	588.53	811.56	998.6	1201.32	1419.81	2357.17
9	713.17	967.77	1299.68	1421.18	1634.21	2787.05
11	770.00	1129.01	1344.34	1691.19	2033.16	3171.8
13	908.49	1332.11	1566.67	1893.44	2304.9	3909.13
19	1331.77	1714.65	2348.26	2479.57	3104.75	5090.28

the symplectic integrator. Moreover, this model will be exploited to optimize the DLO manipulation process. Also, an extension to the method will be applied to multi-branch DLOs, like the pre-assembled wiring harness utilized in the aerospace and automotive industries.

Acknowledgements

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.

References

[1] Jayender, J., Patel, R. V., and Nikumb, S., 2009. "Robot-assisted active catheter insertion: Algorithms and experiments". *The International Journal of Robotics Research,* **28**(9), pp. 1101–1117.

[2] Masey, R. J. M., Gray, J. O., Dodd, T. J., and Caldwell, D. G., 2010. "Guidelines for the design of low-cost robots for the food industry". *Industrial Robot: An International Journal.*

[3] Ramisa, A., Alenya, G., Moreno-Noguer, F., and Torras, C., 2012. "Using depth and appearance features for informed robot grasping of highly wrinkled clothes". In

Table 6. Effects of changing the model parameters $(n_u \& n_s)$ on MATLAB execution time for solving the DLO model employing the symplectic integrator in case of applying an external force.

n_u	n_s					
	51	**81**	**101**	**121**	**151**	**251**
5	489.98	677.93	888.11	1004.88	1254.07	2015.19
7	538.59	881.56	1007.06	1214.88	1554.83	2379.88
9	643.93	970.64	1124.15	1405.01	1675.14	2970.31
11	717.9	1101.62	1385.67	1692.81	2006.02	3140.75
13	874.72	1291.78	1568.77	1886.01	2313.28	3903.67
19	1386.62	1826.59	2102.17	2667.17	2990.85	5035.5

Table 7. Effects of changing the model parameters $(n_u \& n_s)$ on C++ execution time for solving the DLO model employing the symplectic integrator without applying an external force.

n_u	n_s					
	51	**81**	**101**	**121**	**151**	**251**
5	155.28	232.5	284.52	339.25	416.16	673.25
7	197.56	295.17	363.86	435.05	529.7	861.73
9	245.61	369.37	455.4	523.42	652.07	1065.55
11	286.957	436.69	534.16	623.1	765.75	1271.19
13	319.786	489.52	601.8	713.24	887.17	1437.82
19	450.95	690.64	840.81	990.89	1236.14	2021.42

Table 8. Effects of changing the model parameters $(n_u \& n_s)$ on C++ execution time for solving the DLO model employing the symplectic integrator in case of applying an external force.

n_u	n_s					
	51	**81**	**101**	**121**	**151**	**251**
5	154.18	231.83	283.43	335.97	414.29	680.41
7	195.2	295.25	361.43	427.65	530.54	863.73
9	238.25	358.64	442.07	521.7	645.43	1060.26
11	278.72	426.39	523.94	617.38	764.78	1253.72
13	319.12	488.51	600.33	709.03	889.24	1436.29
19	442.82	677.01	835.82	988.3	1228.88	2023.9

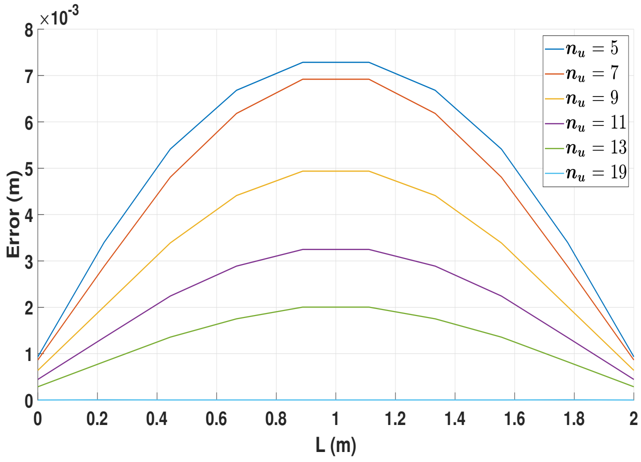


Fig. 12. Evolution of the error over DLO length at different n_u points.

2012 IEEE International Conference on Robotics and Automation, IEEE, pp. 1703–1708.

[4] Shah, A., Blumberg, L., and Shah, J., 2018. "Planning for manipulation of interlinked deformable linear objects with applications to aircraft assembly". *IEEE Transactions on Automation Science and Engineering,* **15**(4), pp. 1823–1838.

[5] Jiang, X., Koo, K.-m., Kikuchi, K., Konno, A., and Uchiyama, M., 2011. "Robotized assembly of a wire harness in a car production line". *Advanced Robotics,* **25**(3-4), pp. 473–489.

[6] Hermansson, T., Bohlin, R., Carlson, J. S., and Söderberg, R., 2013. "Automatic assembly path planning for wiring harness installations". *Journal of manufacturing systems,* **32**(3), pp. 417–422.

[7] Sanchez, J., Corrales, J.-A., Bouzgarrou, B.-C., and Mezouar, Y., 2018. "Robotic manipulation and sensing of deformable objects in domestic and industrial applications: a survey". *The International Journal of Robotics Research,* **37**(7), pp. 688–716.

[8] Moll, M., and Kavraki, L. E., 2006. "Path planning for deformable linear objects". *IEEE Transactions on Robotics,* **22**(4), pp. 625–636.

[9] Huang, S. H., Pan, J., Mulcaire, G., and Abbeel, P., 2015. "Leveraging appearance priors in non-rigid registration, with application to manipulation of deformable objects". In 2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), IEEE, pp. 878–885.

[10] Lui, W. H., and Saxena, A., 2013. "Tangled: Learning to untangle ropes with rgb-d perception". In 2013 IEEE/RSJ International Conference on Intelligent Robots and Systems, IEEE, pp. 837–844.

[11] Wang, W., Berenson, D., and Balkcom, D., 2015. "An online method for tight-tolerance insertion tasks for string and rope". In 2015 IEEE International Conference on Robotics and Automation (ICRA), IEEE, pp. 2488–2495.

[12] Rambow, M., Schauß, T., Buss, M., and Hirche, S., 2012. "Autonomous manipulation of deformable objects based on teleoperated demonstrations". In 2012 IEEE/RSJ International Conference on Intelligent Robots and Systems, IEEE, pp. 2809–2814.

[13] Jiménez, P., 2012. "Survey on model-based manipulation planning of deformable objects". *Robotics and computer-integrated manufacturing,* **28**(2), pp. 154–163.

[14] Hu, W., Wang, Z., Zhao, Y., and Deng, Z., 2020. "Symmetry breaking of infinite-dimensional dynamic system". *Applied Mathematics Letters,* **103**, p. 106207.

[15] Hu, W., Ye, J., and Deng, Z., 2020. "Internal resonance of a flexible beam in a spatial tethered system". *Journal of Sound and Vibration,* **475**, p. 115286.

[16] Hu, W., Xu, M., Song, J., Gao, Q., and Deng, Z., 2021. "Coupling dynamic behaviors of flexible stretching hub-beam system". *Mechanical Systems and Signal Processing,* **151**, p. 107389.

[17] Hu, W., Zhang, C., and Deng, Z., 2020. "Vibration

and elastic wave propagation in spatial flexible damping panel attached to four special springs". *Communications in Nonlinear Science and Numerical Simulation,* **84**, p. 105199.

[18] Hu, W., Yu, L., and Deng, Z., 2020. "Minimum control energy of spatial beam with assumed attitude adjustment target". *Acta Mechanica Solida Sinica,* **33**(1), pp. 51–60.

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

[20] Servin, M., and Lacoursiere, C., 2008. "Rigid body cable for virtual environments". *IEEE Transactions on Visualization and Computer Graphics,* **14**(4), pp. 783–796.

[21] Linn, J., and Dreßler, K., 2017. "Discrete cosserat rod models based on the difference geometry of framed curves for interactive simulation of flexible cables". In *Math for the Digital Factory*. Springer, pp. 289–319.

[22] Valentini, P. P., and Pennestrì, E., 2011. "Modeling elastic beams using dynamic splines". *Multibody system dynamics,* **25**(3), pp. 271–284.

[23] Greco, L., and Cuomo, M., 2013. "B-spline interpolation of kirchhoff-love space rods". *Computer Methods in Applied Mechanics and Engineering,* **256**, pp. 251–269.

[24] Lv, N., Liu, J., Xia, H., Ma, J., and Yang, X., 2020. "A review of techniques for modeling flexible cables". *Computer-Aided Design,* **122**, p. 102826.

[25] Theetten, A., Grisoni, L., Andriot, C., and Barsky, B., 2008. "Geometrically exact dynamic splines". *Computer-Aided Design,* **40**(1), pp. 35–48.

[26] Theetten, A., Grisoni, L., Duriez, C., and Merlhiot, X., 2007. "Quasi-dynamic splines". In Proceedings of the 2007 ACM symposium on Solid and physical modeling, pp. 409–414.

[27] Palli, G., 2020. "Model-based manipulation of deformable linear objects by multivariate dynamic splines". In 2020 IEEE Conference on Industrial Cyberphysical Systems (ICPS), Vol. 1, IEEE, pp. 520–525.

[28] Kinoshita, H., Yoshida, H., and Nakai, H., 1991. "Symplectic integrators and their application to dynamical astronomy". *Celestial Mechanics and Dynamical Astronomy,* **50**, pp. 59–71.

[29] Feng, K., 1984. "On difference schemes and symplectic geometry". In Proceedings of the 5th international symposium on differential geometry and differential equations.

[30] Sanz-Serna, J., 1988. "Runge-kutta schemes for hamiltonian systems". *BIT Numerical Mathematics,* **28**(4), pp. 877–883.

[31] Hong, J., Huang, C., and Wang, X., 2018. "Symplectic runge–kutta methods for hamiltonian systems driven by gaussian rough paths". *Applied Numerical Mathematics,* **129**, pp. 120–136.

[32] Hu, W., Yin, T., Zheng, W., and Deng, Z., 2020. "Symplectic analysis on orbit-attitude coupling dynamic problem of spatial rigid rod". *Journal of Vibration and Control,* **26**(17-18), pp. 1614–1624.

[33] Hu, W., Deng, Z., Han, S., and Zhang, W., 2013. "Generalized multi-symplectic integrators for a class of hamiltonian nonlinear wave pdes". *Journal of Computational Physics,* **235**, pp. 394–406.

[34] Forest, E., 1987. Canonical integrators as tracking codes (or how to integrate perturbation theory with tracking). Tech. rep., Lawrence Berkeley Lab.

[35] Forest, E., and Ruth, R. D., 1990. "Fourth-order symplectic integration". *Physica D: Nonlinear Phenomena,* **43**(1), pp. 105–117.

[36] Neri, F., 1987. "Lie algebras and canonical integration". *Dept. of Physics, University of Maryland*.

[37] De Boor, C., 2001. "A practical guide to splines; rev. ed., ser. applied mathematical sciences".

[38] Nocent, O., and Remion, Y., 2001. "Continuous deformation energy for dynamic material splines subject to finite displacements". In *Computer Animation and Simulation 2001*. Springer, pp. 87–97.

[39] Hamill, P., 2014. *A student's guide to Lagrangians and Hamiltonians*. Cambridge University Press.

[40] Zhai, W., 2020. "Numerical method and computer simulation for analysis of vehicle–track coupled dynamics". In *Vehicle–track coupled dynamics*. Springer, pp. 203–229.