



**HAL**  
open science

## Dynamics of continuum and soft robots: a strain parametrization based approach

Frédéric Boyer, Vincent Lebastard, Fabien Candelier, Federico Renda

► **To cite this version:**

Frédéric Boyer, Vincent Lebastard, Fabien Candelier, Federico Renda. Dynamics of continuum and soft robots: a strain parametrization based approach. IEEE Transactions on Robotics, 2020, 10.1109/TRO.2020.3036618 . hal-02318617

**HAL Id: hal-02318617**

**<https://hal.science/hal-02318617>**

Submitted on 18 Oct 2019

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# Forward dynamics of continuum and soft robots: a strain parametrization based approach

Frederic Boyer, Vincent Lebastard, Fabien Candelier, Federico Renda.

**Abstract**—In this article we propose a new solution to the forward dynamics of Cosserat beams with in perspective, its application to continuum and soft robotics manipulation and locomotion. In contrast to usual approaches, it is based on the non-linear parametrization of the beam shape by its strain fields and their discretization on a functional basis of strain modes. While remaining geometrically exact, the approach provides a minimal set of ordinary differential equations in the usual Lagrange matrix form that can be solved with standard explicit time-integrators. Inspired from rigid robotics, the calculation of the matrices of the Lagrange model is performed with a continuous inverse Newton-Euler algorithm. The approach is tested on several numerical benches of non-linear structural statics, as well as further examples illustrating its capabilities for dynamics.

## I. INTRODUCTION

A nonlinear Cosserat beam is defined by stacking along a single material dimension a continuous set of rigid cross sections of infinitesimal thickness [1]. In contrast to the so-called Euler-Bernoulli, Rayleigh or Timoshenko models of strength of materials, in this non-linear theory, whose the most general instantiation is due to Reissner [2], the cross sections can be subject to finite magnitude transformations [3]. With the recent progresses of continuum and soft robotics [4], this model is increasingly used today. In medical robotics, it is an efficient modelling tool for designing and analysing new concepts of highly redundant needles and endoscopes consisting of a passive elongated organ internally actuated through different technologies [5], [6], [7],[8]. In bio-robotics, it has been used for modelling locomotion of continuum elongated animals as snakes, worms and fish with the aim of understanding the secrets of their performance and to reproduce them on bio-inspired artifacts [9],[10]. More recently, taking inspiration from hydrostats as the octopus, the model of Cosserat beams has attracted attention of the young community of soft robotics for modelling arms made of soft materials as silicon, actuated with cables, pressurized chambers or electro-active materials [11], [12]. There exists a powerful numerical method specifically designed to simulate the Reissner beam theory, namely the Geometrically-Exact Finite Elements Method (GE-FEM) initiated by J.C. Simo in the 80s [13], and developed by

successors as [14], [15], [16], [17]. However, with one exception [18], it seems that this numerical method does not spread in the robotics community. One of the major reasons of this is probably that the GE-FEM is heavy, and practically difficult to implement for researchers not involved in non-linear structural dynamics. Furthermore, even in this research field, it is not conventional since it requires to cope with the differential geometry of the Lie group  $SO(3)$  [13], its Stieffel fibration  $S^2 \times SO(2)$  [16], or directly  $SE(3)$  [17]. Another obstacle to the penetration of the GE-FEM in robotics, is due to the fact that the beam configuration being parameterized with the absolute pose of its cross-sectional frames, when imposing internal kinematic constraints (as the inextensibility condition), or external ones (e.g. those of a joint between two beams), the dynamics turn out to be ruled by stiff Differential-Algebraic-Equations (DAEs), which are not easy to numerically solve, and even more difficult to exploit for control purposes. Alternatively to the FEM, the oceanic engineering community have addressed in the 90s the forward dynamics of undersea cables using the shearing-free Cosserat model of Kirchhoff rods [19], [20]. In this other context, implicit finite difference schemes in time are used to set the dynamics of the cable in the form of a nonlinear Two Point Boundary Value Problem (TPBVP) in space, that can be changed into a set of algebraic equations with a space finite difference scheme, and numerically addressed with a finite-difference solver in space-time [19], or directly solved at each time-step with a shooting method [20], [21]. In both cases, the nonlinear TPBVP is linearized to be iteratively solved at each time step of a simulation. Recently, this shooting approach has been improved and applied to the real time simulation of several design of continuum robots [22]. The computational graphics community has also extensively used the Cosserat model, with the aim of designing fast interactive simulations of rods able to be bent and twisted into complex shapes with knots. In this context, authors of [23] have proposed an original spatial discretization of a Cosserat (Kirchhoff) rod by integral averaging its rotation field along edges connecting a set of discrete vertices. Coupled with discrete parallel transport of frames, this alternative approach is geometrically exact and has been shown recently to be a physical realistic solution for addressing several forward and inverse dynamics of Cosserat rods [24].

F. Boyer and V. Lebastard are with the LS2N lab, Institut Mines Telecom Atlantique, 44307 Nantes, France. e-mail: frederic.boyer@imt-atlantique.fr, vincent.lebastard@imt-atlantique.fr.

F. Candelier is with Aix-Marseille University (AMU), IUSTI lab France. e-mail: fabien.candelier@univ-amu.fr.

F. Renda is with the Khalifa University Center for Autonomous Robotics Institute and the Healthcare Engineering Innovation Center, Khalifa University of Science and Technology, Abu Dhabi, UAE. e-mail: federico.renda@ku.ac.ae.

Despite their respective interest, being based on the same absolute parametrization, all these approaches do not provide dynamic models in the form of a minimal set of ODEs, as those governing the joint-parameterized dynamics

of our usual manipulators [25]. This is in contrast to the so-called Piecewise-Constant-Curvature (PCC) parametrization, an approach which has rapidly spread in robotics [26]. In this approach, the beam is first partitioned into a serial connection of pieces along which the curvatures are assumed to be constant and used as configuration parameters for the robot. Originally introduced with a further torsional component in [27], this kind of parametrization has been recently extended in the framework of Cosserat beams for soft robots through the Piecewise-Constant-Strain (PCS) parametrization [28]. In this approach, the configuration of a Cosserat beam is no more described with the absolute poses of its cross-sectional frames, but rather with the relative poses between them, or in short, with its strain fields. Being constant, these fields can be used as parameters of a set of finite transformations similar to the joint dependent inter-body transformations of a rigid manipulator. Owing to its relative nature, such a parametrization allows setting the dynamic model of soft robots in the usual form of rigid ones, i.e., it provides some minimal sets of ODEs suited to the design of model-based controllers as well as to the use of standard explicit time-integrators for simulation [29]. Moreover, the approach allows modelling more easily the effects of actuation and proprioceptive sensing.

In the article here presented, we push forward the strain based parametrization of the PCS approach, by considering that the strains are no more constant along some pieces, but rather discretized on a truncated basis of strain functions according to the Ritz method [30]. To assess the feasibility of this new parametrization, and in particular its ability to represent the statics and dynamics of non-linear Cosserat beams, we here apply it to the entire domain of a single beam which is kinematically free or clamped at each of its ends, and subject to any set of external and internal forces. While opening promising perspectives for the multi-beam and multi-section case, this first milestone can already be used in any application where a continuum robot can be represented by an actuated single beam. In particular, the internal constitutive law can encompass an activation component able to model the internal actuation of a soft arm [31],[22], or the muscular activity of an elongated animal, as this is expected in biomechanics of fish swimming or snake creeping [32]. As far as we know, the approach is also new for non-linear structural dynamics, where the discretisation of passive non-linear beams always concerns the positional (and rotational in the GE case) fields, and not the strain ones. It can easily capture in an explicit form, all the sub-models of the full Reissner model, as the shearing free Kirchhoff model, or that of inextensible Kirchhoff beams. For future application to bioinspired locomotion, we consider the general case of a beam free at its both ends, whose overall rigid motion is parameterized by the pose of its root frame. In this regard, the approach can be seen as a geometrically exact generalisation of the floating frame approach of flexible robots [33], where the root frame stands for the floating frame, and the deformation

are no more linearly parameterized on a basis of vibrational modes, but rather non-linearly with a basis of strain modes. Such a ("modal") reduction is of great interest for robotics since it allows shifting from an infinite dimensional model to a finite dimensional one with few modes, to which one can apply control and optimization technics [25]. As a consequence of this new parametrization, two further difficulties arise. First, in contrast to the PCS [28], the strain fields being not constant along the beam, the kinematics are no more analytically integrable. Second, due to the relative parametrization, all the geometric nonlinearities express through the mass operator which becomes full, highly nonlinear, and difficult to calculate. To address the first issue, the beam kinematics are numerically integrated with a quaternion-based integrator. Regarding the second issue, all the matrices of the inverse Lagrangian dynamic model of the beam are calculated with the Newton-Euler (NE) inverse dynamic algorithm of an equivalent continuum manipulator according to an idea originally proposed in [34] for calculating the Lagrangian model of rigid systems. Using this inverse NE algorithm, the approach, which is at that time, not optimized in terms of computational time, is however very easy to implement.

The article is structured as follows. In section 2, the model of Cosserat beam is reminded, with its usual absolute parametrization. In section 3, we show how this model can be adapted to bioinspired, continuum and soft robotics by adding a model of the actuation to the passive constitutive law of the standard model. In section 4, the strain based parametrization of a Cosserat beam and its discretization on a Ritz basis are presented. Based on this relative parametrization, the forward dynamic model of an internally free-free actuated beam is provided in section 5. Its practical numerical computation is performed with a general algorithm presented in section 6. This general algorithm requires using several times the NE inverse dynamic algorithm of an equivalent virtual manipulator here derived in section 7. In section 8, the general algorithm of section 5 is applied to several numerical benches proposed in the past years to validate the GE FEM, as well as to other examples illustrating the approach in dynamics.

## II. MODEL OF COSSERAT BEAMS

For the purpose of concision, we use a few of the Lie group notations of [35], and invite the reader not familiar with them, to refer to Appendix 1. We here consider an hyper-elastic beam subject to finite displacements and small strains. In the Cosserat approach, such a medium is modeled by a continuous set of rigid cross sections stacked along a material line parameterized by a coordinate  $X \in [0, 1]$  which plays the role of a continuous label for the cross sections. To each  $X$ -cross section, a mobile cross-sectional frame  $\mathcal{F}(X) = (O, t_1, t_2, t_3)(X)$  is attached, where  $O(X)$  and  $t_1(X)$  coincide with the center of the cross section and its unit normal vector, respectively. In this context, the

configuration of any cross section is defined by the action of an element of  $g \in SE(3)$  on a unique inertial frame  $\mathcal{F}_s$ . Hence, the beam configuration space is naturally defined as:

$$\mathcal{C} = \{g : X \in [0, 1] \mapsto g(X) \in SE(3)\}, \quad (1)$$

which stands for a functional space of curves in  $SE(3)$ . Throughout the article, partial derivatives  $\partial./\partial X$  and  $\partial./\partial t$  are indicated by a "prime" and a "dot", respectively. The field  $g$  depending on both  $X$  and  $t$ , its space-time variations can be entirely described by the two vector fields  $\eta$  and  $\xi$  from  $[0, 1]$  to  $se(3) \cong \mathbb{R}^6$  (see Appendix 1 for notations):

$$\eta = (g^{-1}\dot{g})^\vee, \quad \xi = (g^{-1}g')^\vee, \quad (2)$$

where  $\eta = (\Omega_1, \Omega_2, \Omega_3, V_1, V_2, V_3)^T = (\Omega^T, V^T)^T$  stands for the field of the velocity twists of the cross-sections in their mobile frames, while  $\xi = (K_1, K_2, K_3, \Gamma_1, \Gamma_2, \Gamma_3) = (K^T, \Gamma^T)^T$  is the exact geometrical counterpart of  $\eta$  when replacing  $t$  by  $X$ . By a slight abuse of terminology, this second field of twist will be often named the strain field, since any set of strain tensors has to depend on the current configuration  $g$  through this field only [3]. For instance, if  $g_o$  defines the reference configuration of the beam in which its internal strain energy is zero, a strain field is simply provided by  $\xi - \xi_o$  with  $\xi_o = (g_o^{-1}g'_o)^\vee$ . The velocity and strain fields  $\eta$  and  $\xi$  have their dual counterparts in  $se(3)^* \cong \mathbb{R}^6$ . Denoted  $\Sigma = (\Upsilon_1, \Upsilon_2, \Upsilon_3, P_1, P_2, P_3)^T = (\Upsilon^T, P^T)^T$  and  $\Lambda = (C_1, C_2, C_3, N_1, N_2, N_3)^T = (C^T, N^T)^T$ , these two further fields of wrench model the cross-sectional kinetic momenta and stress along the beam respectively. In the small strains assumption, these two pairs of dual fields are related by the (generalized) constitutive relations:

$$\Sigma = \mathcal{M}\eta, \quad \Lambda = \mathcal{H}(\xi - \xi_o), \quad (3)$$

where  $\mathcal{M} = \text{diag}(\rho I_1, \rho I_2, \rho I_3, \rho A, \rho A, \rho A)$  and  $\mathcal{H} = \text{diag}(GI_1, EI_2, EI_3, EA, GA, GA)$  are two tensors fields in  $se(3)^* \otimes se(3) \cong \mathbb{R}^6 \otimes \mathbb{R}^6$ , which stand for the cross-sectional inertia and Hooke tensors along the beam. Applying Hamilton principle in this geometric context, provides the partial differential equation (PDE) of a Cosserat beam subject to a density of state-dependent external wrench  $\bar{F}$  on  $]0, 1[$  and two tip external wrenches  $F_0$  and  $F_1$  at  $X = 0$  and  $1$  respectively [3]:

$$\dot{\Sigma} - ad_\eta^T \Sigma = \Lambda' - ad_\xi^T \Lambda + \bar{F}, \quad (4)$$

with boundary conditions at  $X = 0$ :

$$\Lambda(0) = -F_0, \quad \text{or } g(0) = g_0, \quad (5)$$

and at  $X = 1$ :

$$\Lambda(1) = F_1, \quad \text{or } g(1) = g_1, \quad (6)$$

depending whether, we impose the motion (through some known tip transformations  $g_0$  and  $g_1$ ), or the force at the tips. From (3), introducing  $\Sigma = \mathcal{M}\eta$  into (4) provides a PDE in acceleration  $\dot{\eta}$  which governs the time-evolution of  $\eta$  and needs to be supplemented with the reconstruction equation (a kinematic model):

$$\dot{g} = g\dot{\eta}. \quad (7)$$

Finally, the Cosserat formulation (3-7) stands for the model of Reissner beams [2], which is at the basis of the geometrically exact finite element method of [13]. In this case, (4) is replaced by its equivalent weak-form of virtual works, which once spatially discretized in finite-elements, is solved at each step of a time-loop with respect to a set of nodal velocities. Then the beam configuration is updated by using a discrete version of the reconstruction equation (7), using the exponential map [13].

**Remark 1:** In all the subsequent developments, we will consider that the internal beam kinematics which define the motions between cross sections can be constrained by some restrictions. For instance, if we neglect the transverse shearing and the extensibility, one has  $\Gamma = (1, 0, 0)^T$ , and the beam kinematics can be reduced from the 6 internal dofs of a Reissner beam to the 3 internal dofs of those of a Kirchhoff beam [36]. To generically handle this kind of restriction, we can use the following decomposition of the field  $\xi$ :

$$\xi = B\xi_a + \bar{B}\xi_c, \quad (8)$$

where for any  $n_a \leq 6$ ,  $\xi_a$  (respectively  $\xi_c$ ) is the  $n_a \times 1$  vector field of the free strains allowed by the beam kinematics (respectively, the  $(6 - n_a) \times 1$  vector field of prohibited strains), while  $B$  and  $\bar{B}$  are two complementary selection matrices of 1 and 0 such that  $B^T B = 1$ ,  $B^T \bar{B} = 0$  and  $\bar{B}^T \bar{B} = 1$ . Going further into details,  $\xi_a$ , with the subscript "a" meaning "allowed", is a field of kinematic unknowns governed by the dynamics of the beam, while  $\xi_c$ , with "c" for "constrained", is defined by some prescribed constants with respect to time. For instance, for an inextensible Kirchhoff beam, we have:

$$\xi_a = (K_1, K_2, K_3)^T, \quad \xi_c = (1, 0, 0)^T, \quad (9)$$

$$B = \begin{pmatrix} 1_{3 \times 3} \\ 0_{3 \times 3} \end{pmatrix}, \quad \bar{B} = \begin{pmatrix} 0_{3 \times 3} \\ 1_{3 \times 3} \end{pmatrix}.$$

Finally, the decomposition (8) has its counterpart on the stress:

$$\Lambda = B\Lambda_a + \bar{B}\Lambda_c, \quad (10)$$

where  $\Lambda_a$  is the field of stress components governed by the reduced constitutive law:

$$\Lambda_a = B^T \Lambda = \mathcal{H}_a(\xi_a - \xi_{ao}), \quad (11)$$

with  $\mathcal{H}_a = B^T \mathcal{H} B$ , the matrix of the reduced Hooke coefficients, and  $\xi_{ao}$  the value of  $\xi_a$  in the reference configuration. Finally, in (10),  $\Lambda_c$  stands for a set of Lagrange multipliers fields (internal reaction forces and torques), in charge of imposing the internal constraints  $\bar{B}^T \xi = \xi_c$ .

### III. APPLICATION TO CONTINUUM AND SOFT ROBOTICS

Though being originally motivated by the study of passive beams in finite deformations, the above formulation can be easily adapted to model several systems relevant to bio-inspired and continuum robotics. Indeed, in several fields of robotics, continuum robots can be considered as

internally actuated Cosserat beams [31],[22]. Going further into details, modelling the body of the robot by the above model, and assuming that the inertia of the actuators are negligible with respect to those of this actuated body, the model of such an internally actuated Cosserat beam can be easily deduced from the above passive one, by replacing the reduced constitutive law (11), by the active one:

$$\Lambda_a = \Lambda_{ad}(t) + \mathcal{D}_a \dot{\xi}_a + \mathcal{H}_a(\xi_a - \xi_{ao}), \quad (12)$$

where  $\Lambda_{ad}$  now stands for the autonomous time-dependent internal forces and/or torques produced by the actuation (in all the paper, the subscript "d" denotes some imposed forces defined by prescribed time-laws), while  $\mathcal{H}_a$ , and  $\mathcal{D}_a = B^T DB$  model the viscoelastic properties of the actuated passive beam, all being compatible with the allowed internal beam kinematics. We now consider the model of section 2 with the active constitutive law (12), and illustrate how it can be used for addressing several issues in robotics.

Imposing  $\mathcal{D}_a = \mathcal{H}_a = 0$  in (12), provides the dynamic model of a continuum multibody system constituted of an infinite number of bodies of infinitesimal length, namely the cross sections. Such a model has been used in [9] for solving the continuum inverse dynamics of hyper-redundant fish robots. In the same context, (12) more generally provides a simplified model of the muscles inside an elongated animal, with  $\Lambda_{ad}$  standing for the exogenous rhythmic stress field governed by a central pattern generator, while  $(\mathcal{D}_a, \mathcal{H}_a)$  can be used for modelling a certain damping and stiffness of the muscles that are here mounted in parallel with the actuation [37]. Note that in this first context, the Cosserat model is used as an idealization of the original continuum animals from which bio-robotics takes its inspiration.

The same law can be used in other fields, such as medical or soft robotics, where roboticists have designed elongated structures which are internally actuated by different types of technologies like pressurized chambers [38], or routine cables [31]. In this latter context, the passive organ of the robot (e.g., a flexible rod along which discrete disks are arranged, a piece of silicon through which cables are passed...) can be modelled by a Cosserat beam (or a serial connection of several of them), i.e., with the above model, where in (12),  $(\mathcal{D}_a, \mathcal{H}_a)$  encode the viscoelastic properties of the flexible passive organ, while  $\Lambda_{ad}$  captures the effects of the actuators on the system. In practical applications, the model of  $\Lambda_{ad}$  has to be derived on a case-by-case basis. A possible derivation consists in shifting the actuation forces from the physical space, where they are primarily exerted, to the space of allowed internal strains  $\xi_a$ , by using the kinematic-invariance of their internal virtual works.

To illustrate such a derivation, we consider the case of a cable driven soft arm as in [31]. This system is modelled by an inextensible Kirchhoff beam actuated by a set of frictionless cables of negligible inertia and extension indexed  $i =$

$1, 2, \dots, N$ , each of them being pulled in a controlled manner. Such cables are attached to the tip cross section  $X = 1$  and routed through any other  $X \in [0, 1[$ , in such a way that in any beam cross sectional frame  $\mathcal{F}(X)$ , the  $i^{\text{th}}$  cable crosses the section at a position:  $D_i(X) = (0, D_{i,Y}, D_{i,Z})(X)$ . In these conditions, each cable has for positional field inside the beam:  $X \mapsto \varphi_i(X) = r + RD_i$ , and is able to transmit along its unit tangent  $\varphi'_i / \|\varphi'_i\|$ , a tension  $t \mapsto T_i(t) \in \mathbb{R}^+$  only. The cable length and its variation are defined by:

$$l_i = \int_0^1 \|\varphi'_i\| dX, \quad \delta l_i = \int_0^1 \frac{\delta \varphi_i^T \varphi'_i}{\|\varphi'_i\|} dX. \quad (13)$$

Introducing the expressions of  $\varphi_i = r + RD_i$  into (13), one can easily show that any variation  $\delta l_i$ , can be expressed in terms of the beam shape variations  $\delta \xi = (\delta K^T, \delta \Gamma^T)^T$ , as:

$$\delta l_i = \int_0^1 \frac{(\Gamma + K \times D_i + D'_i)^T (\delta \Gamma + \delta K \times D_i)}{\|\Gamma + K \times D_i + D'_i\|} dX. \quad (14)$$

The above identity defines the Jacobian from the kinematic space of fields  $\delta \xi_a = B^T \delta \xi$  to the discrete set of  $\delta l_i$ s (which stands for the starting kinematic space of the cables). Therefore, using duality, i.e., invoking the kinematic-invariance of internal virtual works, we have for  $N$  cables:

$$\delta W = \sum_{i=1}^N \delta l_i T_i = \int_0^1 \delta \xi_a^T \Lambda_{ad} dX. \quad (15)$$

Introducing (14) into (15) provides by simple identification, the expected vector of internal wrenches exerted on the Kirchhoff beam by a set of  $N$  cables:

$$\begin{aligned} \Lambda_{ad}(X, t) &= \sum_{i=1}^N B^T \begin{pmatrix} D_i \times \Gamma_{c,i} \\ \Gamma_{c,i} \end{pmatrix} \frac{T_i(t)}{\|\Gamma_{c,i}\|} \\ &= \sum_{i=1}^N \left( \frac{D_i(X) \times \Gamma_{c,i}}{\|\Gamma_{c,i}\|} \right) T_i(t). \end{aligned} \quad (16)$$

where we used the  $B$ -matrix of an inextensible Kirchhoff beam (9), with notations  $\Gamma_{c,i} = E_1 + K \times D_i + D'_i$ ,  $E_1 = (1, 0, 0)^T$ , and where the subscript "c" here means "cable". Note that when  $K$  is constant over the beam, this field becomes uniform and has been obtained in [31] by other means. Finally, in all the subsequent developments we consider the model of section 2 with the full constitutive law (12), and refer to it as that of an internally actuated Cosserat beam, or more concisely as that of a continuum robot.

#### IV. STRAIN PARAMETRIZATION OF A COSSERAT BEAM

In the original Cosserat model, the configuration space of a beam is defined as (1), i.e. each configuration of the beam is defined as the field of homogeneous transformations  $g$  parameterizing the pose of its cross-sectional frames with respect to a unique inertial frame. In [9], we introduced an alternative definition of the configuration space of a Cosserat beam which is based on the following simple remark. If one knows at each instant the pose  $g(0) \triangleq g_0$  of the root cross-sectional frame  $\mathcal{F}(0) \triangleq \mathcal{F}_0$  w.r.t. the inertial frame,

as well as the field of spatial twist  $\xi$  along the beam length, one can then easily reconstruct the field of  $g$  along it, by forward integrating (i.e. from  $X = 0$  to 1), the continuous geometric model:

$$g' = g\hat{\xi}. \quad (17)$$

with initial condition  $g(0) = g_0$ . Therefore, one can parametrize the Cosserat beam configuration by  $(g_0, \xi)$ , where remind that  $\xi$  is a field of  $6 \times 1$  vectors in  $se(3) \cong \mathbb{R}^6$ . This leads to the second definition of configuration space of a floating beam:

$$\mathcal{C} = SE(3) \times \mathbb{S}, \quad (18)$$

where:

$$\mathbb{S} = \{\xi : X \in [0, 1] \mapsto \xi(X) \in \mathbb{R}^6\}. \quad (19)$$

Geometrically, such a space defines a principal fiber bundle  $SE(3) \times \mathbb{S}$ , where the fiber  $SE(3)$  here stands for the configuration space of  $\mathcal{F}_0$ , while the space  $\mathbb{S}$  is the configuration space of its shapes, or "shape space". The beam being a continuous system,  $\mathbb{S}$  is a functional space, i.e. an infinite dimensional space of  $X$ -parameterized curves in  $\mathbb{R}^6$ . Practically, the reconstruction of the beam configuration in space can be easily performed by numerically integrating (17) as follows. Parameterizing the rotation matrix  $R$  of  $g = (R, r)$ , with a unit quaternion  $Q \in \mathbb{R}^4$ , one can integrate (17) in the alternative form:

$$\begin{pmatrix} Q \\ r \end{pmatrix}' = \begin{pmatrix} (2\|Q\|)^{-1}A(R(Q)K)Q \\ R(Q)\Gamma \end{pmatrix}, \quad (20)$$

starting from  $Q_0(R_0)$  and  $r_0$ , both deduced from  $g_0 = (R_0, r_0)$ . In (20), we introduced the usual operator :

$$A(K) = \begin{pmatrix} 0 & -K_1 & -K_2 & -K_3 \\ +K_1 & 0 & -K_3 & +K_2 \\ +K_2 & +K_3 & 0 & -K_1 \\ +K_3 & -K_2 & +K_1 & 0 \end{pmatrix}, \quad (21)$$

as well as  $R(Q) \in SO(3)$  and  $Q(R) \in \mathbb{R}^4$ , which stand for the standard quaternion parametrization of rotations and its inverse map, respectively [35]. Now, referring to remark 1, for any Cosserat beam kinematics, one can adopt (18) as definition of the configuration space, with now:

$$\mathbb{S} = \{\xi_a : X \in [0, 1] \mapsto \xi_a(X) \in \mathbb{R}^{n_a}\}, \quad (22)$$

which includes the general case of Reissner beams for which  $n_a = 6$ . Finally, for the purpose of numerical resolution, we now discretize the field  $\xi_a$  as:

$$\xi_a(X, t) = \xi_{ao}(X) + \Phi(X)q(t), \quad (23)$$

where  $\Phi = (\Phi_1, \Phi_2, \dots, \Phi_n)$  defines a  $n_a \times n$  matrix of  $n$  basis functions or "shape functions" since they parameterize the shape space  $\mathbb{S}$  of (18) with the  $n \times 1$  vector  $q$  of strain generalized coordinates. With this further reduction, the configuration space of a Cosserat beam simply reads:

$$\mathcal{C} = SE(3) \times \mathbb{R}^n. \quad (24)$$

This configuration space, which is still a principal fiber bundle, is well known of rigid robotics locomotion [39]. In

this other context the finite dimensional shape space of (24) stands for the coordinates that parameterise the  $n$  joints connecting a set of  $n + 1$  rigid bodies  $\mathcal{B}_0, \mathcal{B}_1 \dots \mathcal{B}_n$ , while the fiber  $SE(3)$  is the configuration space of an arbitrarily distinguished reference body  $\mathcal{B}_0$ .

**Remark 2:** In (23), remind that  $\xi_{ao}(X)$  is the  $X$ -rate of variation of  $g$  on a reference configuration of the beam. As a result, if the beam is straight in its reference configuration, we have  $\xi_{ao} = B^T \xi_o$  with  $\xi_o = (0, 0, 0, 1, 0, 0)^T$ . In this case the only non-zero component of  $\xi_{ao}$  is brought by an allowed stretching field, as this is the case for an extensible Kirchhoff beam. More generally, if the reference configuration is curved, we need to shift from Cosserat beams to arches and  $(K_{2o}, K_{3o})$  will introduce further nonzero components in the time-independent field  $\xi_{ao}$ .

## V. DYNAMICS OF A CONTINUUM ROBOT PARAMETERIZED BY STRAINS

We can now build on the above geometric picture and seek for a dynamic model governing the configuration  $(g_0, q)$  along time. To that end, one can reconsider the weak form of virtual works in the most general case where  $\Lambda_a$  is defined by an active constitutive law of the type (12). A straightforward application of this principle in the configuration space (24) provides the dynamic model of a continuum robot in Lagrangian form:

$$\begin{pmatrix} 0 \\ Q_{ad} \end{pmatrix} = \begin{pmatrix} M_0 & M_{0\epsilon} \\ M_{\epsilon 0} & M_{\epsilon\epsilon} \end{pmatrix} \begin{pmatrix} \dot{\eta}_0 \\ \dot{q} \end{pmatrix} + \begin{pmatrix} F_v(q, \dot{q}, \eta_0) \\ Q_v(q, \dot{q}, \eta_0) \end{pmatrix} + \begin{pmatrix} F_c(q, g_0) \\ Q_c(q, g_0) \end{pmatrix} + \begin{pmatrix} 0 \\ K_{\epsilon\epsilon}q + D_{\epsilon\epsilon}\dot{q} \end{pmatrix}, \quad (25)$$

which needs to be supplemented with the reconstruction equation  $\dot{g}_0 = g_0 \hat{\eta}_0$ , i.e., a kinematic model allowing to recover the time evolution of  $g_0$  from that of  $\eta_0$  governed by the dynamic model (25). Due to our definition (24) of the configuration space, all these matrices are organized in sub-blocks and the top row of the above matrix equality represents a balance of wrench (i.e.  $6 \times 1$  vectors) w.r.t. to the reference frame  $\mathcal{F}_0$ , while the bottom one, stands for a balance of  $n \times 1$  generalized forces in the shape space (here parameterized by the strain coordinates of  $q$ ). Going into details, in the right hand side of (25), we find from left to right, the matrix of generalized inertia of the Cosserat beam, the vector of its accelerations, that of all velocity-dependent forces, which include the Coriolis and centrifugal forces as well as some eventual velocity-dependent external forces as those exerted on a swimming robot. The next vector stands for that of configuration-dependent external forces as the gravitational ones, or a set of imposed time-varying forces appearing first in the two concentrated wrenches  $F_0, F_1$  and/or the wrench density  $\bar{F}$ . Next, we find the generalized

restoring and damping forces, where we have introduced the matrices of generalised stiffness and damping:

$$K_{\epsilon\epsilon} = \int_0^1 \Phi^T \mathcal{H}_a \Phi dX, \quad D_{\epsilon\epsilon} = \int_0^1 \Phi^T \mathcal{D}_a \Phi dX. \quad (26)$$

Finally, on the left hand side of (25),  $Q_{ad}$  stands for the vector of generalised forces of internal actuation:

$$Q_{ad}(t) = - \int_0^1 \Phi(X)^T \Lambda_{ad}(X, t) dX. \quad (27)$$

**Remark 3:** The model (25) is enough general to encompass several cases in nonlinear structural dynamics, biomechanics and soft robotics. For instance, when the robot is a soft manipulator, we just have to remove the d.o.f.s of the floating frame  $\mathcal{F}_0$  (i.e., the top row of (25)). If one wants to consider the ideal case of a continuum rigid robot internally actuated, one has just to remove the restoring and damping forces. On the other hand, if one wants to consider the case of a passive Cosserat beam, it suffices to remove the activation forces  $Q_{ad}$ . Moreover, if in this latter case the stiffness matrix is zero, the same model can be used to describe a three dimensional continuum rigid chain swinging in gravity. More generally, using the full model (25) allows describing the locomotion dynamics of an elongated animal as a fish or a snake actuated by the continuous muscular model (12). Finally, for specific technological systems as those recently developed for soft robotics and medical needles [40],[5], the right hand side of (25) remains valid, while the modelling effort is confined to the calculation of  $\Lambda_{ad}$  in  $Q_{ad}$ . For instance, if one reconsiders the example of the cable driven inextensible Kirchhoff beam of section 3, introducing (16) into (27) allows detailing the left hand side of (25) in the form:

$$Q_{ad} = - \sum_{i=1}^N \left( \int_0^1 \Phi^T (D_i \times \Gamma_{c,i}) dX \right) T_i(t) = LT \quad (28)$$

where  $L(q)$  denotes the  $n \times N$  actuation matrix, and  $T = (T_1, T_2, \dots, T_N)^T$  is the vector of control variables.

**Remark 4:** The strains being not subject to boundary conditions of geometric type (these conditions are imposed to the field of transformations  $g$ ), the shape functions of (23) are not required to satisfy any constraints at the boundaries of  $[0, 1]$ . Different functional basis can be used to discretize the strain fields as long as the discretization is set in the form (23). For instance, in case where  $n_a = 1$  the basis which comes immediately to mind is a polynomial basis of monomials of increasing order and we shall have:

$$\Phi(X) = (1, X, X^2, \dots, X^{n-1}) \triangleq P_n(X). \quad (29)$$

Note that the PCS parametrization of [28] is a special case of (29) in which all the non zero powers of  $X$  are removed. Continuing with polynomials one could give more structure to them and replace the above basis of monomials by an orthogonal basis of Tchebychev polynomials. In this case which will be numerically tested, the stiffness

matrix (26) is changed into the identity matrix  $1_{n_a \times n_a}$ . Finally, when  $n_a > 1$ , i.e. if one considers several allowed strains (possibly actuated), one has just to repeat the above scalar discretization for each of the strain components. For instance, if  $\xi_a = (K_1, K_2, K_3)^T$ , one can use (29) for each of the three components and obtain the form (23) with the block-diagonal matrix:

$$\Phi(X) = \text{diag}(P_{k_1}(X), P_{k_2}(X), P_{k_3}(X)), \quad (30)$$

where  $k_1 + k_2 + k_3 = n$ , i.e.  $q = (q_1, q_2, \dots, q_n)$ , with  $(q_1, \dots, q_{k_1})$ ,  $(q_{k_1+1}, \dots, q_{k_1+k_2})$  and  $(q_{k_1+k_2+1}, \dots, q_n)$  representing the torsional, and the curvature coordinates (in the planes  $(t_1, t_2)$  and  $(t_1, t_3)$ ) respectively.

**Remark 5:** Thanks to the strain parametrization, the stiffness and damping matrices of (26) as well as the actuation matrix of (28), are particularly simple and the two first can be calculated once for all outside the time-loop of a simulation. On the other hand, the explicit calculation of the other matrices is much more complex and in fact analytically unfeasible. Going further into details, when shifting from the  $g$ -parametrisation of section 2, to the strain parametrisation, the geometric nonlinearities are transferred from the model of the restoring forces to that of the inertial forces. As a result, while our stiffness matrices are easily computable, the mass matrices and velocity dependent forces are highly non-linear and a priori difficult to calculate. In the next section we propose a simple algorithm for calculating these matrices numerically. Inspired from past works in rigid robotics [34], the principle of this algorithm consists in using an inverse Newton-Euler dynamic algorithm of the system to numerically reconstruct all the matrices of its Lagrangian model. In a further step, we will see how such an inverse NE algorithm can be derived.

## VI. PRACTICAL CALCULATION OF THE FORWARD DYNAMIC MODEL OF CONTINUUM ROBOTS

To practically calculate all the matrices of (25), it is convenient, for reasons that will be detailed soon, to consider a fully actuated virtual rigid continuum manipulator associated to our free-free Cosserat beam. By "virtual manipulator", we mean that the reference section  $X = 0$  is no more considered as being free but rather virtually actuated with an external wrench  $F_0$  exerted on it. Moreover, this manipulator is said "rigid" and "fully actuated" since the field  $\xi_a$  is directly actuated with no internal damping nor stiffness. In these conditions, (25) is simply changed into:

$$\begin{pmatrix} F_0 \\ Q_a \end{pmatrix} = \begin{pmatrix} \mathcal{M}_0 & M_{0\epsilon} \\ M_{\epsilon 0} & M_{\epsilon\epsilon} \end{pmatrix} \begin{pmatrix} \dot{\eta}_0 \\ \dot{q} \end{pmatrix} + \begin{pmatrix} F_v(q, \dot{q}, \eta_0) \\ Q_v(q, \dot{q}, \eta_0) \end{pmatrix} + \begin{pmatrix} F_c(q, g_0) \\ Q_c(q, g_0) \end{pmatrix}. \quad (31)$$

In short, (31) defines the Lagrangian model of the virtual continuum manipulator associated with the original beam, and except the damping and restoring forces, all the matrices of (25) are handled by (31). As any Lagrangian model,

(31) is an inverse dynamic model which is explicit w.r.t. the accelerations. Alternatively, it is well known from rigid robotics [41] that the same dynamics can be described in an implicit form, thanks to an inverse Newton-Euler (NE) algorithm as that proposed by Luh and Walker [42] or by Featherstone [43] for rigid multibody systems. In the present case, such a Newton-Euler algorithm provides for any state  $(g_0, \eta_0, q, \dot{q})$  of the manipulator, the outputs  $F_0$  and  $Q_a$  that accelerate the manipulator with  $(\dot{\eta}_0, \ddot{q})$ . Therefore, this inverse NE algorithm defines an input-output map, here named *IDM* (for Inverse Dynamics Model), of the general form:

$$\begin{pmatrix} F_0 \\ Q_a \end{pmatrix} = IDM(q, \dot{q}, \ddot{q}, g_0, \eta_0, \dot{\eta}_0). \quad (32)$$

Now, according to [34], if one has in hand such an *IDM*, by feeding it with some specific known inputs, it becomes easy to calculate all the matrices of the Lagrangian model (25) at each instant of a simulation time-loop. Going further into details, let us define  $\delta_i \in \mathbb{R}^6$  (respectively  $\delta_\alpha \in \mathbb{R}^n$ ) as a vector with all entries equal to zero except the  $i^{th}$  one (respectively, the  $\alpha^{th}$  one), which is equal to 1. Then, using the definitions of  $\delta_i, \delta_\alpha$ , simple identifications of (31) and (32) provides us with the following algorithm, where for any matrix  $A$ ,  $A^{(k)}$  denotes its  $k^{th}$  column (with  $k = i = 1, 2, \dots, 6$ , or  $k = \alpha = 1, 2, \dots, n$ ):

- Step 1: Calculate:

$$\begin{pmatrix} F_c \\ Q_c \end{pmatrix} = IDM(q, 0, 0, g_0, 0, 0) \quad (33)$$

- Step 2: Calculate:

$$\begin{pmatrix} F_v \\ Q_v \end{pmatrix} = IDM(q, \dot{q}, 0, g_0, \eta_0, 0) - \begin{pmatrix} F_c \\ Q_c \end{pmatrix}. \quad (34)$$

- Step 3: For  $i = 1, 2, \dots, 6$ , calculate:

$$\begin{aligned} \begin{pmatrix} \mathcal{M}_0^{(i)} \\ M_{\epsilon 0}^{(i)} \end{pmatrix} &= \begin{pmatrix} \mathcal{M}_0 \delta_i \\ M_{\epsilon 0} \delta_i \end{pmatrix} \\ &= IDM(q, 0, 0, g_0, 0, \delta_i) - \begin{pmatrix} F_c \\ Q_c \end{pmatrix}. \end{aligned} \quad (35)$$

- Step 4: For  $\alpha = 1, 2, \dots, n$ , calculate:

$$\begin{aligned} \begin{pmatrix} M_{0\epsilon}^{(\alpha)} \\ M_{\epsilon\epsilon}^{(\alpha)} \end{pmatrix} &= \begin{pmatrix} M_{0\epsilon} \delta_\alpha \\ M_{\epsilon\epsilon} \delta_\alpha \end{pmatrix} \\ &= IDM(q, 0, \delta_\alpha, g_0, 0, 0) - \begin{pmatrix} F_c \\ Q_c \end{pmatrix}. \end{aligned} \quad (36)$$

Finally, when all the matrices of (31) are computed, one can simply add (26) and (27) to build the model (25), invert the mass matrix with the Cholesky algorithm, and deduce the forward dynamics of the original actuated free-free Cosserat beam in the explicit form:

$$\frac{d}{dt} \begin{pmatrix} q \\ g_0 \\ \eta_0 \\ \dot{q} \end{pmatrix} = \begin{pmatrix} \dot{q} \\ g_0 \dot{\eta}_0 \\ \left( \begin{pmatrix} \mathcal{M}_0 & M_{0\epsilon} \\ M_{\epsilon 0} & M_{\epsilon\epsilon} \end{pmatrix}^{-1} \begin{pmatrix} -F_{tot} \\ -Q_{tot} \end{pmatrix} \right) \end{pmatrix}, \quad (37)$$

where we used notations  $F_{tot} = F_v + F_c$  and  $Q_{tot} = Q_v + Q_c + K_{\epsilon\epsilon}q + D_{\epsilon\epsilon}\dot{q} - Q_{ad}$ . In the above system, the second equation (from top to bottom), is a reconstruction equation which can be integrated with the quaternion-based integration scheme (20) where  $t$  replaces  $X$ . Finally, time-integrating numerically (37) also provides the time evolution of  $q$  from which the configuration of all the cross-sectional frames can be reconstructed at any instant by integrating (20) with  $\xi$  given by (8,23).

**Remark 6:** Note that if we have to deal with a soft manipulator, i.e., an actuated Cosserat beam clamped in  $X = 0$  and free in  $X = 1$ , referring to remark 3, its dynamics can be more simply written in the well known Lagrangian form :

$$Q_a = M_{\epsilon\epsilon}(q)\ddot{q} + Q_v(q, \dot{q}) + Q_c(q), \quad (38)$$

and the corresponding *IDM* takes the reduced form  $Q_a = IDM(q, \dot{q}, \ddot{q})$ . This reduced *IDM* which was originally introduced in [34] to calculate the matrices of (38) for rigid manipulators is no more than (32) where  $g_0 = 1, \eta_0 = \dot{\eta}_0 = 0$ . This confirms on the NE side, that the case of a free-free actuated beam includes that of soft actuated manipulators, and for this reason in all the following we shall continue to consider this more general context.

**Remark 7:** Finally, the above forward dynamic algorithm is conditioned by the obtention of a NE inverse dynamic algorithm *IDM* of a continuum manipulator in the form (32). Such an algorithm is presented in the next section. It's derivation is based on the generalisation of the Newton-Euler formulation from rigid multibody systems to continuum ones.

## VII. INVERSE NE DYNAMICS OF CONTINUUM MANIPULATORS

In [9], a continuum NE inverse algorithm has been proposed for solving the inverse dynamics of a free-free Cosserat beam modelling a continuum hyper-redundant locomotor. This algorithm has been extensively used for the study of bio-inspired swimming [44]. In this section, we remind it in the case of a continuum manipulator. In a second step, we use this continuous *IDM* to design the (discrete) *IDM* required by the forward dynamics algorithm of section 6.

### A. Statement of the problem

We address the inverse dynamic problem of a fully actuated Cosserat beam, i.e., a beam actuated by a field of internal wrench  $\Lambda_a$  and an external wrench  $F_0$ . This problem can be stated as follows. Imposing any two sets  $(g_0, \eta_0, \dot{\eta}_0)$  and  $(\xi_a, \dot{\xi}_a, \ddot{\xi}_a)$ , find the field  $\Lambda_a$  and the wrench  $F_0$  that ensure them. This formally defines the continuous *IDM* as the input-output map:

$$\begin{pmatrix} F_0 \\ \Lambda_a \end{pmatrix} = IDM(\xi_a, \dot{\xi}_a, \ddot{\xi}_a, g_0, \eta_0, \dot{\eta}_0), \quad (39)$$



where let us remind that in contrast to (32), the inputs  $\xi_a, \dot{\xi}_a, \ddot{\xi}_a$  and the output  $\Lambda_a$  are fields, i.e., elements of some infinite-dimensional functional spaces.

### B. Continuous inverse NE algorithm

For solving (39), we first remark that introducing  $\Sigma = \mathcal{M}\eta$  into the PDEs (4) gives:

$$\Lambda' = ad_{\xi}^T \Lambda + \mathcal{M}\dot{\eta} - ad_{\eta}^T \mathcal{M}\eta - \bar{F}, \quad (40)$$

which are no more than the NE equations of a rigid continuum robot, where the cross sections stand for the discrete bodies, and the stress  $\Lambda$ , for the interbody wrenches. Furthermore, differentiating twice (17) provides the continuous version of the usual recursions on transformations, velocities and accelerations of the discrete NE model of manipulators [41]. Exploiting this NE structure of the Cosserat model, one can easily derive a continuum version of the computed torque algorithm of rigid manipulators [25]. To that end, it suffices first to forward  $X$ -integrate (from  $X = 0$  to  $X = 1$ ), the continuum model of transformations, velocities and accelerations along the beam:

$$\begin{pmatrix} g \\ \eta \\ \dot{\eta} \end{pmatrix}' = \begin{pmatrix} g\hat{\xi} \\ -ad_{\xi}\eta + B\dot{\xi}_a \\ -ad_{\xi}\dot{\eta} - ad_{B\xi_a}\eta + B\ddot{\xi}_a \end{pmatrix}, \quad (41)$$

which provides all the kinematic fields from the basis, where they are imposed, to the tip. Once all the kinematics are known over the entire beam, one can introduce them into (40), that is  $X$ -integrated backward from the tip  $X = 1$ , where  $\Lambda(1) = F_1$  is imposed, to  $X = 0$ , where  $\Lambda(0)$  is an unknown reaction wrench. This second spatial integration provides the field of stress wrench  $\Lambda$  and finally the expected outputs of (39). This algorithm is summarized as follows and detailed in Appendix 2:

- 1) Forward  $X$ -integrate equations (41) initialized with  $(g_0, \eta_0, \dot{\eta}_0)$ .
- 2) Calculate the state-dependent functions  $F_1, \bar{F}, ad_{\eta}^T(\mathcal{M}\eta)$ .
- 3) Backward  $X$ -integrate (40) with initial condition  $\Lambda(1) = F_1$ .
- 4) Once  $\Lambda$  known along  $[0, 1]$ , calculate the outputs:  $\Lambda_a = B^T \Lambda$  and  $F_0 = -\Lambda(0)$ .

**Remark 8:** The above algorithm is nothing more than a continuous version of the inverse algorithm of Luh [42]. As its discrete homologous, it is structured into similar functional steps, where the 2 (1 forward and 1 backward) recursions on body indices of the discrete case, are replaced by 2 ODEs (1 forward and 1 backward) on the continuous label  $X$ . Therefore, one can easily hybridise the above continuum inverse algorithm with that of [42], in order to produce a more general *IDM* holding for systems connecting several soft and/or rigid bodies in a tree-like topology. To that end,

the forward kinematic ODEs of the above algorithm (step 1) which holds along soft bodies, need to be composed with the usual discrete recursions on transformations, velocities and accelerations along rigid bodies. In the same manner, the backward ODEs on the internal stress wrenches of step 3 are composed with the discrete recursion on inter-body wrenches of the rigid elements. In these compositions, the final conditions provided by integrating an ODE, or iterating a recursion, need first to be changed of frame with one of the discrete operators  $g, Ad_g$  or  $Ad_g^T$ , where  $g$  here denotes a discrete transformation at the junction between two bodies. Once so transformed, these final conditions feed the initial conditions of the next ODE or recursion. Repeating this process from body to body along the topology, provides the expected *IDM* which ensures the continuity of all the fields (transformation, velocities, accelerations and stress) throughout the entire structure.

### C. Discrete inverse NE algorithm

Now that we have a realisation of the continuous *IDM* of a continuum manipulator, it is straightforward to build a discrete version of the same algorithm, which is consistent with our strain discretisation. Indeed, it suffices to impose (23) along with:

$$\dot{\xi}_a(X, t) = \Phi(X)\dot{q}(t), \quad \ddot{\xi}_a(X, t) = \Phi(X)\ddot{q}(t), \quad (42)$$

as inputs of the continuous *IDM*, and to project its output  $\Lambda_a$  on the same functional basis according to:

$$Q_a = - \int_0^1 \Phi(X)^T \Lambda_a(X) dX. \quad (43)$$

This discretisation of the continuous *IDM* input-output map defines the discrete *IDM* of a continuum manipulator (32).

**Remark 9:** Finally, applying the algorithm of section 6. with this discrete *IDM* allows reconstructing all the matrices of the Lagrangian model (25), except those of the generalised restoring and damping ones, whose expressions (26) are indeed enough simple to be added a posteriori (see remark 5). Nevertheless, if one wants to automatically deduce the restoring and damping generalized forces with the reconstruction algorithm, it suffices to replace  $-\Lambda_a$ , by  $-\Lambda_{ad} + \mathcal{H}_a(\xi_a - \xi_{ao}) + \mathcal{D}_a \dot{\xi}_a$  in the continuum *IDM* and to gather the restoring forces (respectively, the damping forces) with the configuration (respectively, the velocity) dependent ones, while  $\Lambda_{ad}$  replaces  $\Lambda_a$  as an output of *IDM*. This output being real if we deal with an internally actuated continuum robot, or fictitious, if we deal with a passive beam.

## VIII. NUMERICAL TESTS

We first address a set of benches used in the past to evaluate the GE-FEM of Reissner and Kirchhoff beams [45],[16]. In these examples, the beam is considered as passive and modelled with the general constitutive law (12)

with no activation, and with  $\mathcal{D}_a = \mu \mathcal{H}_a$ , where  $\mu$  is a coefficient tuning the internal damping. We close this section with the case of a cable driven beam free of external forces.

#### A. Static simulations

In static regime, the BCs of a beam need to impose at least 6 independent geometric constraints at its tips. Moreover, the beam needs to be subject to slow variations of external forces or/and geometric boundary conditions. In this case, the external d.o.f. along with the accelerations (including Coriolis and centrifugal ones) can be removed from the model (25) which turns out to be:

$$K_{\epsilon\epsilon}q = -Q_c(q, t), \quad (44)$$

where  $Q_c$  here represents some generalized external forces which can be directly imposed or indirectly defined as some reaction forces (Lagrange multipliers) forcing a set of slowly moving geometric constraints. In particular,  $t$  in (44) stands for a fictitious numerical time that encodes the incremental variations imposed by these external forces or/and geometric constraints. The static balance (44) defines a set of nonlinear algebraic equations that can be solved with a Newton algorithm. However, this approach can be advantageously circumvented by adding an internal damping  $D_{\epsilon\epsilon}$  artificially chosen, its role being to change the set of algebraic equations (44) into the set of ODEs:

$$D_{\epsilon\epsilon}\dot{q} + K_{\epsilon\epsilon}q = -Q_c(q, t), \quad (45)$$

which can be integrated explicitly with respect to the numerical time  $t$ . Note here that we are only interested in the asymptotic solutions of (45), which are the roots of (44) as soon as  $D_{\epsilon\epsilon}$  is such that the dissipations have enough time to operate between successive incremental excitations. In short, with a good accordance of the numerical damping matrix and the numerical time step  $\Delta t$ , the beam will follow a succession of quasi-static equilibriums leading it to the expected static equilibrium configuration. In this section, (44) and (45) are named the undamped and damped static balances respectively.

1) *Test 1: Clamped beam subject to a concentrated tip torque:* This first test is very usual since it corresponds to one of the only analytical solutions of Cosserat rods in finite deformations. This exact solution was numerically recovered with the GE-FEM in [45],[16]. The beam is clamped in  $X = 0$  and subject to an external imposed concentrated torque at  $X = 1$ . As a result, the forward algorithm has just to calculate  $Q_c$  with (33), in which  $\bar{F} = 0$  and  $F_1 = F_1(t) = (0, 0, C(t), 0, 0, 0)^T$ . This calculation can be performed numerically with the general algorithm. However, in this particular case, and for the purpose of the illustration, one can entirely integrate by hand the undamped static balance (44) as follows. First, using the relation  $ad_{\xi}^T = -Ad_g^T Ad_g^{-T}$  (see Appendix 1), the ODE (40) can be explicitly integrated with  $\bar{F} = \eta = \dot{\eta} = 0$ , which gives:

$$Q_c(t) = - \int_0^1 \Phi^T B^T Ad_g^T dX Ad_{g(1)}^{-T} F_1(t). \quad (46)$$

Second, since the load is planar, one can consider that only the curvature  $K_3$  is excited and take  $B = (0, 0, 1, 0, 0, 0)^T$ . In these conditions, adopting as functional basis (29), the integral (46) can be detailed as:

$$Q_c(t) = - \int_0^1 \Phi^T dXC(t). \quad (47)$$

Which, after integration gives:

$$Q_c(t) = -C(t)(1, 1/2, 1/3, 1/4 \dots 1/n)^T, \quad (48)$$

which defines the right hand member of the static balance (44) (we assume  $\mu = 0$ ). Regarding the left hand side of this balance, it is straightforward to calculate the stiffness matrix with (26) for the strain basis (29), which gives the static balance:

$$EI \begin{pmatrix} 1 & \frac{1}{2} & \frac{1}{3} & \dots & \frac{1}{n} \\ \frac{1}{2} & \frac{1}{3} & \frac{1}{4} & \dots & \frac{1}{n+1} \\ \frac{1}{3} & \frac{1}{4} & \frac{1}{5} & \dots & \frac{1}{n+2} \\ \dots & \dots & \dots & \dots & \dots \\ \frac{1}{n} & \frac{1}{n+1} & \frac{1}{n+2} & \dots & \frac{1}{2n} \end{pmatrix} \begin{pmatrix} q_1 \\ q_2 \\ \vdots \\ q_n \end{pmatrix} = C \begin{pmatrix} 1 \\ \frac{1}{2} \\ \frac{1}{3} \\ \vdots \\ \frac{1}{n} \end{pmatrix}.$$

Now, since the first column of the above  $n \times n$  matrix is equal to  $EI/C$  times the vector of the right hand side, the above system has for solution  $q = (C/EI)(1, 0, 0 \dots 0)^T$ . Moreover, this solution is unique since  $K_{\epsilon\epsilon}$  is invertible as soon as all the basis functions are different. Finally, this simple analysis shows that the strain parametrization allows recovering the exact analytical solution of Euler which stipulates that a beam subject to a tip torque enrolls around itself with a curvature radius of  $EI/C$ .

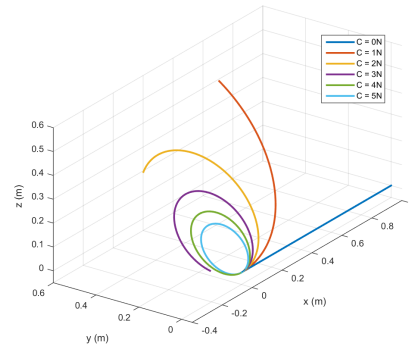


Fig. 1: Beam subject to a concentrated tip torque: Equilibrium configurations for a few loading steps in 3D.

To numerically illustrate this test, we reported in figures 1, 2 and 3 the results of a simulation of the damped balance (45) for a circular cross-sectional beam with diameter  $2R_b = 0.01\text{m}$ , length  $l = 1\text{m}$ , and  $E = 10^8\text{Pa}$ . The simulation is performed by numerically calculating  $Q_c(q)$  with the general algorithm of section 6, fed with  $F_1 = C(t)(e_c^T, 0_{1 \times 3}^T)^T$ , where  $e_c = (0, 1/\sqrt{2}, 1/\sqrt{2})^T$  is the unit vector around which the torque is applied incrementally from 0 to  $C = 5\text{Nm}$  with a step of  $C/k$ , and  $k = 5$  a fixed number of steps. Note that in contrast to the above analytical

solution, the tip torque being applied in an arbitrary tilted plane, we used a strain basis defined by  $\Phi = \text{diag}(P_{k_2}, P_{k_3})$ , with here  $k_2 = k_3 = 5$ . The (numerical) damping matrix is such that  $D_{\epsilon\epsilon} = \mu K_{\epsilon\epsilon}$  with  $\mu = 10^{-3}$ s. The numerical time step between two consecutive increments of load is set to  $\Delta t = 0.02$ s. Figure 1 shows several equilibrium configurations in three-dimensions (3D). Figure 2 illustrates the typical first-order type convergency of  $\dot{K}_2, \dot{K}_3$  along loading steps, while figure 3 displays the evolution of the beam's tip position vs the magnitude of the tip torque. As expected, one recovers the solution of Euler in the tilted symmetry plane where the torque is exerted.

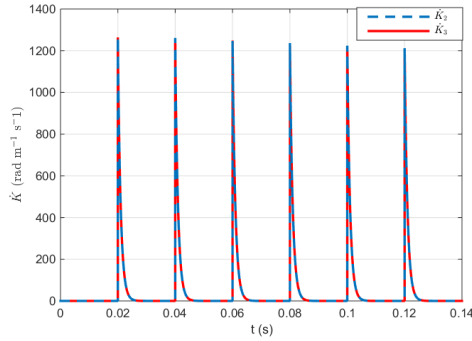


Fig. 2: Beam subject to a concentrated tip torque: Time evolution of  $\dot{K}_2, \dot{K}_3$  for several steps of loading.

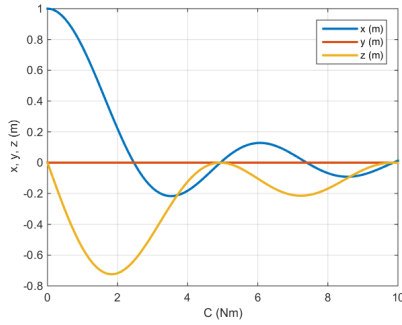


Fig. 3: Beam subject to a concentrated tip torque: Beam's tip position vs tip torque magnitude.

2) *Test 2: Clamped beam subject to a follower tip force:* This test was originally simulated with the GE-FEM applied to Reissner and Kirchhoff beams in [45], and [16] respectively. It consists in imposing to a beam, a follower tip force defined by  $F_1 = (0, 0, 0, 0, F(t), 0)^T$ . In this case, the integral of (46) is no more  $g$ -independent and needs to be calculated numerically with the general algorithm of section 6 in the conditions of (33) where the configuration-dependent external forces are here only due to  $F_1$ . In figures 4 and 5, we reported the results of a simulation of the damped balance (45) obtained with the 5 first monomials of (29) as curvature basis. As in the example of [16], we consider a circular cross-sectional beam with diameter  $2R_b = 0.57$ m, length  $l = 100$ m, and  $E = 4.015 \times 10^8$ Pa.

The simulation is performed by incrementing  $F(t)$  from 0 to 130kN. The numerical damping matrix is such that  $D_{\epsilon\epsilon} = \mu K_{\epsilon\epsilon}$ , with  $\mu = 10^{-3}$ s. The numerical time step between two consecutive increments of load is  $\Delta t = 0.02$ s. Figure 4 displays the equilibrium configurations of the beam for several values of the tip force, while figure 5 shows the evolution of the beam's tip position versus the tip force magnitude. As expected, the results perfectly fit with the GE-FEM of [16].

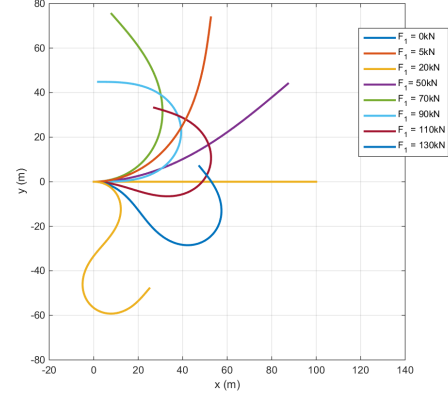


Fig. 4: Beam subject to a follower tip force: Equilibrium configurations for a few loading steps.

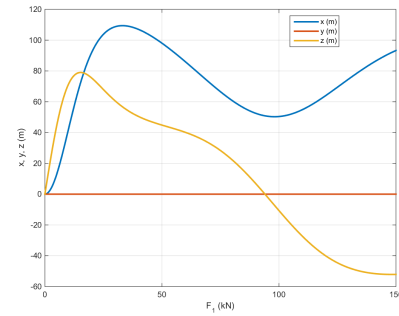


Fig. 5: Beam subject to a follower tip force: Beam's tip position vs tip force magnitude.

3) *Test 3: Clamped-clamped beam subject to imposed tip motions:* We follow with a three-dimensional static test simulated with the GE-FEM of Kirchhoff rods in [16]. This is a key test, since it allows qualifying the couplings between flexion and torsion which are responsible of most of the geometric nonlinearities in statics. It consists in twisting a rod, and then, to push its two tips toward each other. As this can be observed on the electric cables of our computers, the strain energy initially stored in the torsion state is so converted into flexural energy, while the cable forms a loop. To numerically reproduce this example, one considers an extensible Kirchhoff beam with the strain parametrization of remark 2, i.e.:

$$\xi = B(\xi_{ao} + \Phi q) = B((0, 0, 0, 1)^T + \Phi(X)q), \quad (49)$$

with  $\Phi(X) = \text{diag}(P_{k_1}, P_{k_2}, P_{k_3}, P_{k_4})(X)$ . The beam is clamped at its two tips, but while in  $X = 0$ ,  $\mathcal{F}(0)$  is maintained fixed, in a first phase (Phase 1), one imposes  $\mathcal{F}(1)$  to rotate around its first vector  $t_1(1)$  of  $2\pi$ rad, while maintaining its origin fixed. After this first phase, one enters into a second one (Phase 2) along which, while the orientation of  $\mathcal{F}(1)$  is maintained fixed, its origin ( $X = 1$ ) is pushed toward the root  $X = 0$  of a certain displacement. This imposed motion can be modelled by the geometric constraints on transformations:

$$g(1) = g_1(t), \quad (50)$$

where  $g(1)$  depends on  $q$  only, while  $g_1(t)$  is a prescribed tip transformation consistent with the two phases of the test. Time differentiating (50) provides a model of the constraints in the kinematic form:

$$J_1(q)\dot{q} = \eta(1) = \eta_1(t), \quad (51)$$

where  $\eta_1(t) = (g_1^{-1}(t)\dot{g}_1(t))^\vee$  is the prescribed time varying twist imposed at the moving tip of the beam, while  $J_1$  is the  $6 \times n$  linear map encoding how the frame in  $X = 1$  moves (with twist  $\delta\psi(1) = (g(1)^{-1}\delta g(1))^\vee$ ), when one imposes some arbitrary perturbations  $\delta q$  to the strain coordinates. This linear map  $J_1$  can be easily calculated by remarking that the field  $\delta\psi = (g^{-1}\delta g)^\vee$  produced by any  $\delta q$ , is solution of the ODE on  $\eta$  of (41) with  $\eta$  replaced by  $\delta\psi$ , i.e.:

$$\delta\psi' = -ad_\xi\delta\psi + B\Phi\delta q, \quad \delta\psi(0) = 0_{6 \times 1}. \quad (52)$$

Therefore, using  $ad_\xi = Ad_g^{-1}Ad'_g$  (see Appendix 1) allows one to integrate (52) explicitly and to get:

$$J_1 = Ad_{g(1)}^{-1} \int_0^1 Ad_g B\Phi dX, \quad (53)$$

which, according to the duality between forces and velocities, is no more than the transposal of the matrix appearing in (46). Now, if  $g_1(t)$  varies enough slowly along time, the dynamic effects are negligible and the evolution of the beam can be described by the damped dynamics (45), in which the (configuration dependent) external forces consist of:

$$Q_c(q) = Q_g(q) - J_1^T(q)\Lambda_1, \quad (54)$$

where  $Q_g(q)$  is the vector of gravity forces and  $\Lambda_1$  is a wrench of unknown Lagrange multipliers expressed in the cross sectional frame  $X = 1$ , whose role is to force the constraints (51). In these quasi-static conditions, the test can be modelled by the closed system of DAEs:

$$\begin{aligned} D_{\epsilon\epsilon}\dot{q} + K_{\epsilon\epsilon}q &= -Q_g(q) + J_1^T(q)\Lambda_1, \\ J_1(q)\dot{q} &= \eta_1(t). \end{aligned} \quad (55)$$

Instead of using a Newton algorithm, (55) is here addressed with a projective method. To that end, one considers that  $\dot{q}$  is at any time, the sum of a particular and an homogeneous solution of the kinematic constraints, i.e. :

$$\dot{q} = J_1^+(q)\eta_1(t) + H(q)\dot{q}_r, \quad (56)$$

where  $J_1^+$  is the pseudo-inverse of  $J_1$ ,  $H(q)$  is a  $n \times m$  matrix which gathers  $m$  (independent) basis vectors of the kernel of  $J_1$ , while  $\dot{q}_r$  represents a reduced vector of admissible velocities (compatible with the constraints). Projecting (54) onto  $H$ , allows removing  $\Lambda_1$  and to find:

$$\dot{q}_r = -(H^T D_{\epsilon\epsilon} H)^{-1} H^T (K_{\epsilon\epsilon} q + Q_g + D_{\epsilon\epsilon} J_1^+ \eta_1(t)),$$

which once reintroduced into (56), provides the expected ODEs of the beam:

$$\begin{aligned} \dot{q} &= -H(H^T D_{\epsilon\epsilon} H)^{-1} H^T (K_{\epsilon\epsilon} q + Q_g(q)) \\ &+ (1 - H(H^T D_{\epsilon\epsilon} H)^{-1} H^T D_{\epsilon\epsilon}) J_1^+(q)\eta_1(t). \end{aligned} \quad (57)$$

In these equations,  $\eta_1(t)$  can be considered as a kinematic feedback control of the form:

$$\begin{aligned} \eta_1(t) &= T_{SE(3)}(\Psi_1)\dot{\Psi}_1, \quad \text{with: } \dot{\Psi}_1 = k_p\Psi_1, \quad (58) \\ \text{and: } \Psi_1 &= \Psi_1(q, t) = \log_{SE(3)}(g_1^{-1}(t)g(1)), \end{aligned}$$

where  $T_{SE(3)}$  and  $\log_{SE(3)}$  denote the tangent and the inverse of the exponential map  $\Psi \mapsto \exp_{SE(3)}(\Psi)$  in  $SE(3)$  [17], while  $k_p$  is a diagonal matrix of proportional gains. When applied to (57) with  $g_1(t)$  incremented step by step along the two phases of the test, the control (58) ensures (50). Note that at each new  $q$ , provided by the time-integration of  $\dot{q}$ , one has to  $X$ -integrate forward the continuous geometric model  $g' = g(B\xi_{ao} + \Phi(X)q)^\wedge$  from  $g(0) = 1_{4 \times 4}$ , to find the new  $g(1)$  required by  $\Psi_1$  in (58). Finally, the algorithm of section 6, is only used to calculate  $Q_g$ , with (33) in which  $\bar{F} = (0_{1 \times 3}, (R^T(0, 0, \gamma_g)^T)^T)^T$  and  $\gamma_g = 9.81\text{ms}^{-2}$ .

**Remark 10:** Note that this algorithm only requires to compute  $\log_{SE(3)}$  and  $T_{SE(3)}$ . In contrast, the alternative Newton-algorithm, would require the further operators:

$$T_{SE(3)}^{-1}, \quad \text{and: } \frac{\partial}{\partial \Upsilon} (T_{SE(3)}^{-T}(\Upsilon)\Lambda_1) \in \mathbb{R}^6 \times \mathbb{R}^6.$$

In figure 6, some snapshots of the static equilibria are reported for several values of  $g_1(t)$ . The beam considered is the cross-sectional one of [16] with for diameter 0.0176m, length  $l = 20\text{m}$ ,  $E = 2.1 \times 10^{11}\text{Pa}$ ,  $G = 4.1 \times 10^7\text{Pa}$  and volume mass  $\rho = 2.04 \times 10^8\text{kgm}^{-3}$ . This simulation is carried out with Tchebychev polynomials, namely 5 for each curvature, 3 for torsion, and 2 for stretching ( $k_1 = 3$ ,  $k_2 = k_3 = 5$ ,  $k_4 = 2$ ). Other simulation parameters are  $\mu = 10^{-3}\text{s}$ ,  $k_p = 10^2 \text{diag}(20, 20, 20, 1, 1, 1)$ . With  $E_1 = (1, 0, 0)^T$ , the prescribed tip motion is defined by  $g_1(t) = (\exp_{SO(3)}(\theta_1(t)\hat{E}_1), E_1)$ , for Phase 1, and  $g_1(t) = (1_{3 \times 3}, x_1(t)E_1)$ , for Phase 2. In the two phases,  $\theta_1(t)$  (respectively,  $x_1(t)$ ) is incremented of  $2\pi/k$  (respect., of  $-1.5/k$ ) from 0 to  $2\pi$  (respect., from  $l$  to  $-0.5l$ ), with  $k = 18$  (respect.,  $k = 15$ ), the number of tip pose increments. The numerical time-step between two consecutive increments of tip transformation is  $\Delta t = 0.1\text{s}$ . As expected the torsional energy is converted into flexural one, while the snapshots fit very well (e.g. discrepancies on the torsional field  $K_1$  do not exceed 0.1%) with those given by the GE-FEM of [16].

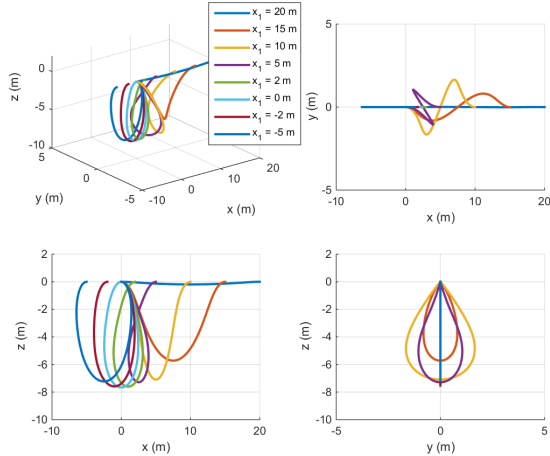


Fig. 6: Flexion-torsion test: Evolution of the beam's configuration for several values of  $g_1(t)$ : (top-left) in 3D, (top-right, bottom left and right) in planes  $x - y$ ,  $x - z$ ,  $y - z$  respectively.

### B. Cantilever Kirchhoff beam actuated with cables

In this section, we consider a cable driven Kirchhoff beam as evoked in the example of section 3. The beam is embedded in  $X = 0$ , free of any external force over its entire length, and internally actuated with one cable. To simulate the effect of imposing a given tension  $T$  along the cable, we need to integrate the damped static balance:

$$D_{\epsilon\epsilon}\dot{q} + K_{\epsilon\epsilon}q = Q_{ad}(q, t), \quad (59)$$

with  $Q_{ad}(q, t) = L(q)T(t)$  the vector of generalized internal actuation forces generated by the cable tension  $T \in \mathbb{R}^+$  and  $L$  defined by (28). Note that simulating (59) does not require using the algorithm of section 6. Indeed, owing to the strain-based parametrization, these matrices are easily computable from (26) and (28). Below, we reported the numerical solutions of (59) obtained with different routines of the cable defined by  $X \mapsto D(X) = (0, D_Y(X), D_Z(X))^T$ , with  $D_Y$  and  $D_Z$  taken as indicated in the table I. These routines, noted parallel, convergent, cross, spiral and parallel-truncated, are defined when the beam is in its reference straight configuration. Figures 7(a,b,c,d) display several beam configurations obtained with  $T = 1, 2, 3, 4, 5\text{N}$  for each routine, except the parallel routine for which one obtains the same deformations as in figure 1 with  $C(t) = (R_b/2)T(t)$ . The simulated beam is that of test 1 but one hundred times softer ( $E = 10^6\text{Pa}$ ). All the deformations are planar except those of the spiral routine whose figure 7.c illustrates one perspective view. All simulations are carried out with 5 modes for each strain field, except in figure 7.d, i.e. for the parallel-truncated routine, for which we represented the results obtained with 5 and 10 modes. In fact except for the parallel-truncated routine, the Ritz discretisation has converged before 5 modes (the deformations do not change significantly by adding

Routine	$D_Y$	$D_Z$	Figure
Parallel	$\sqrt{2}R_b/4$	$\sqrt{2}R_b/4$	1
Convergent	$R_b(1 - X)/2$	0	7.a
Cross	$R_b(1 - 2X)/2$	0	7.b
Spiral	$\sin(2\pi X)$	$\cos(2\pi X)$	7.c
Parallel-truncated	$D_Y = R_b/2, X \in [0, 1/2]$ $D_Y = 0, X \in ]1/2, 1]$	0	7.d

TABLE I: Several routines of a one cable driven beam.

further modes), while due to the discontinuity, the parallel-truncated routine needs ten modes to converge.

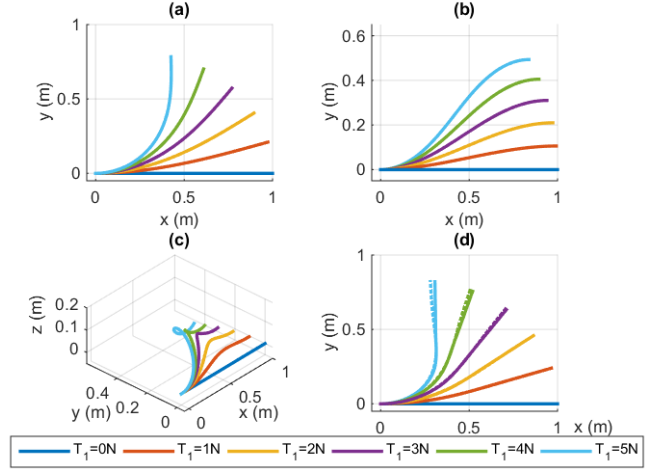


Fig. 7: Beam internally actuated with one cable. Configurations obtained by increasing the tension  $T$  for different routines: (a) Convergent. (b) Cross. (c) Spiral. (d) Parallel-truncated with 5 (dotted lines), and 10 modes (solid lines).

### C. Dynamic simulations

In all subsequent simulations we use the standard ODE45 integrator of Matlab, which allows one to time-integrate non-stiff ODEs.

1) *Test 4: Clamped beam rolled up and released:* In this first test, we start from the static equilibrium of the test 1, i.e., we first apply a concentrated fixed torque of magnitude  $C$  at the tip of a clamped beam. Then, this load is removed instantaneously at  $t = 0$ , and the beam freely moves with no gravity according to the dynamics:

$$M_{\epsilon\epsilon}(q)\ddot{q} + Q_v(q, \dot{q}) + Q_c(q) + D_{\epsilon\epsilon}\dot{q} + K_{\epsilon\epsilon}q = 0, \quad (60)$$

which are deduced from (25) in which the internal activation forces as well as the external dofs are removed (see remark 3 and 7). Except for restoring and damping forces, all the matrices of this model are computed with the full algorithm of section 6. The simulated beam is that of test 1 with volume mass  $\rho = 10^3\text{kgm}^{-3}$  and  $\mu = 10^{-3}\text{s}$ . In figures 8(a,b,c), we reported a set of snapshots plotted each  $\Delta t = 0.12\text{s}$  over the first second of simulation for  $C = 0.8, 2.3, 5.4\text{Nm}$ , while figure 9 shows the time evolution of the coordinates of the beam tip along time. In these figures, the strain basis is a Tchebychev polynomial basis with dimension  $n = 5$ .

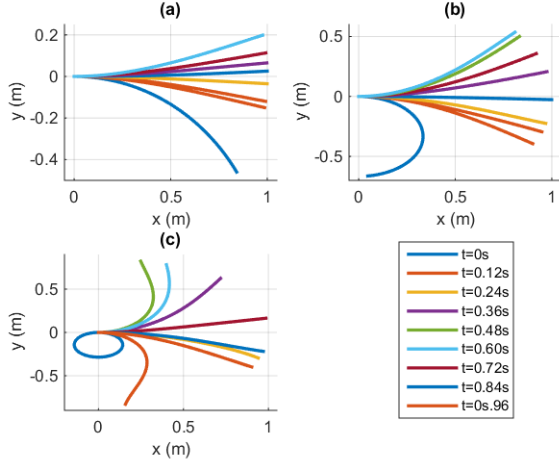


Fig. 8: Clamped beam released after bending with a tip torque  $C$ : Snapshots every 0.12s over the first second of simulation, when  $C = 0.8$  (a), 2.3 (b) and 5.4Nm (c).

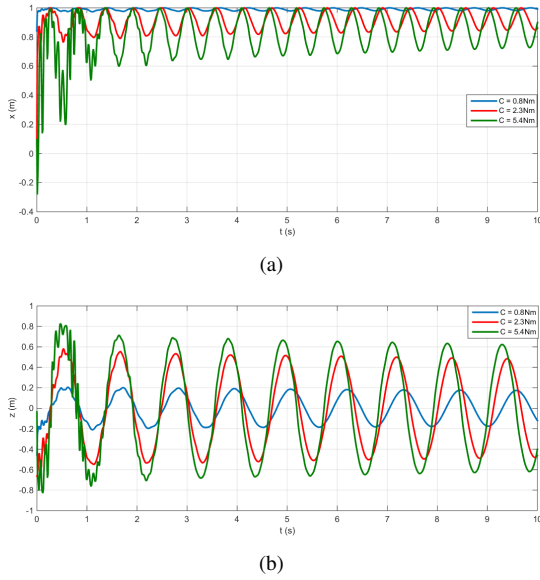


Fig. 9: Clamped beam bent with a tip torque  $C = 0.8, 2.3, 5.4$  Nm and released: Time evolution of its tip abscissa (a), and tip ordinate (b).

To illustrate the convergence of the Ritz decomposition, we use the following error between  $n$  and  $n + 1$  modes:

$$\epsilon(n + 1/n) = \frac{1}{t_f} \int_0^{t_f} \int_0^1 (K_2^{(n+1)} - K_2^{(n)})^2 dX dt, \quad (61)$$

where  $K_2^{(n)}$  stands for  $K_2$  approximated with  $n$  strain functions, while  $t_f$  is the duration of the simulation. Applying (61) to the 3 simulations, shows that  $\epsilon(2/1)$  is always high, while increasing  $n$  makes  $\epsilon(n + 1/n)$  rapidly drop up to be lower than 1% of  $\epsilon(2/1)$ . This rate of convergence decreases when the deformations increase. In short, convergence needs to increase  $n$  with the amplitude of deformations, while  $n = 5$  is enough for three tests. This is confirmed by

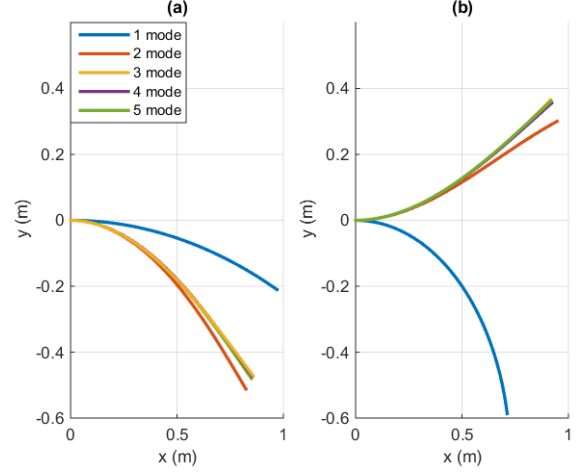


Fig. 10: Beam bent with  $C = 2.3$  Nm and released: Snapshots at  $t = 1$  s (a) and 1.5s (b) with  $n = 1, 2, 3, 4, 5$  modes.

figure 10, which shows 2 representative snapshots with  $n = 1, 2, 3, 4, 5$  modes for  $C = 2.3$  Nm.

2) *Test 5: Three-dimensional continuum pendulum swinging in gravity*: In this second dynamic test, we consider a string whose root  $X = 0$  is clamped to a moving basis while its opposite tip is equipped with a lumped mass  $m_1$  rigidly fixed to it. The string is modelled by an inextensible and non twistable Kirchhoff beam with  $EI_2 = EI_3 = 0$ . According to remark 10, the presence of the tip mass needs to feed the backward ODE (40) of the *IDM* with  $\Lambda(1) = F_1 = -\mathcal{M}_1 \dot{\eta}_1 + F_{in,1} + F_{ext,1}$ , where  $\mathcal{M}_1$  and  $F_{in,1} + F_{ext,1}$  are the  $6 \times 6$  inertia matrix of the tip body and the  $6 \times 1$  wrench of inertia and external forces applied to it, both expressed in  $\mathcal{F}(1)$ . With a lumped mass we have  $\mathcal{M}_1 = -m_1 \text{diag}(0, 0, 0, 1, 1, 1)$ , and  $F_{in,1} + F_{ext,1} = m_1(0_{1 \times 3}, (R^T(1)(\gamma_g, 0, 0)^T)^T - (\Omega_1 \times V_1)^T)^T$ . Finally, this *IDM* can be used to calculate all the matrices of the Lagrangian model of the system, which in this case, reads:

$$\begin{aligned} M_{\epsilon\epsilon}(q)\ddot{q} + Q_v(q, \dot{q}, \eta_0(t)) + Q_c(q) + D_{\epsilon\epsilon}\dot{q} + K_{\epsilon\epsilon}q \\ = -M_{\epsilon 0}(q)\dot{\eta}_0(t), \end{aligned} \quad (62)$$

where  $t \mapsto (\eta_0, \dot{\eta}_0)(t)$  here represent the time-evolution of the velocity and acceleration of the root  $X = 0$  that excite the pendulum. The simulated string has a mass per unit length  $\rho A = 7.85 \times 10^{-5} \text{kgm}^{-1}$  and a length  $l = 1$  m. The tip mass is  $m_1 = 0.03$  kg. In figures 11, 12 and 13, we reported simulation results obtained with  $g_0(t) = (1_{3 \times 3}, r_0(t))$ , and a  $r_0(t)$  defined by a Lissajous curve  $r_0^T(t) = (0, a \sin(\omega t), a \sin(\omega t + \varphi))$ , with  $a = 0.15$  m,  $\omega = 2\pi \text{rad.s}^{-1}$ , and  $\varphi = \pi/2$ . In these conditions,  $\eta_0(t) = (0_{1 \times 3}, \dot{r}_0^T(t))^T$  and  $\dot{\eta}_0(t) = (0_{1 \times 3}, \ddot{r}_0^T(t))^T$ . We used 3 Tchebychev polynomials for  $K_2$  and  $K_3$ , which are the only strains allowed by the internal kinematics. Figure 11 displays several snapshots drawn every  $\Delta t = 0.15$  s. In figure 12, the trajectories of the two tip cross sections are plotted in the  $y-z$  plane, while in figure 13 the coordinates of the tip section are plotted along time.

field governed by (12) with an activation stress of the form:

$$X \in [0, 1] : \Lambda_{ad}(X, t) = \begin{pmatrix} f_r(t)f_w^{(2)}(X, t) \\ f_r(t)f_w^{(3)}(X, t) \end{pmatrix}. \quad (63)$$

Referring to anterior works on swimming [46], [47], the law of each internal torque field is built with two space-time functions. The first, is a sinusoid ramp time-function noted  $f_r$  of the form:

$$\begin{aligned} t \in [0, t_i] : f_r(t) &= 0, \\ t \in [t_i, t_f] : f_r(t) &= \frac{t - t_i}{t_f - t_i} - \frac{1}{2\pi} \sin\left(2\pi \frac{t - t_i}{t_f - t_i}\right), \\ t \geq t_f : f_r(t) &= 1, \end{aligned} \quad (64)$$

which is twice continuously differentiable and thereby guarantees smooth time transitions from 0 to 1. The second function, is a space-time wave function  $f_w^{(i)}$ ,  $i = 2, 3$ , defined  $\forall(X, t) \in [0, 1] \times [0, t_f]$  by:

$$f_w^{(i)}(X, t) = a_i \sin\left[2\pi \left(\frac{X}{\lambda^{(i)}} - \frac{t}{T^{(i)}}\right)\right], \quad (65)$$

in which  $\lambda^{(i)}$  is the wave length,  $T^{(i)}$  is the period, and  $a_i$  is the (constant) amplitude envelop for each of the two curvature components  $i = 2, 3$ . In these conditions, we need to consider the general dynamics (25) whose all matrices are calculated by the algorithm of section 6, except  $Q_{ad}$  which is given by (27) with (63). The simulated beam has the geometry and viscoelastic properties of that of test 4. In figures 14, 15 we reported the results of a simulation carried out with 3 Tchebychev polynomials for each of the curvature fields. The internal law (63) is applied over 2s with  $t_i = 0s$ ,  $t_f = 0.25s$ ,  $(T^{(2)}, T^{(3)}) = (1, 0.25)s$ ,  $(\lambda^{(2)}, \lambda^{(3)}) = (1, 0.5)m$ ,  $a_2 = a_3 = 0.2Nm$ . In figure 14, snapshots every 0.3s are drawn, while figures 15 display the two actuated curvature fields  $\xi_a = (K_2, K_3)^T$  along the beam at the same instants. For assessing the consistency of the simulation, we computed the kinetic wrench  $(\sigma_{tot}^T, p_{tot}^T)^T = \int_0^1 Ad_g^{-T}(\mathcal{M}\eta)dX$ , with  $p_{tot}$  and  $\sigma_{tot}$  the total kinetic resultant and momentum of the beam in the inertial frame. As expected, owing to the absence of external forces, these 6 first integrals are nearly preserved along the simulation ( $\|\sigma_{tot}\|$  in  $kg.m^2/s$ , and the norm of the center mass position in m, are both lower than  $10^{-3}$ ). Note that these conservation properties need to be fulfilled for future application to bio-inspired locomotion.

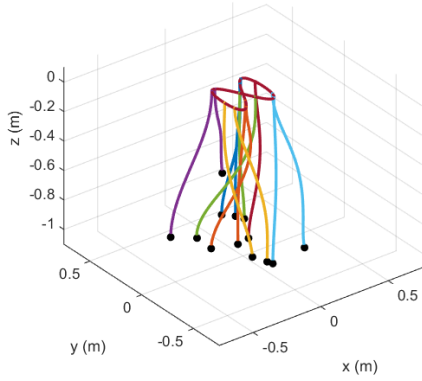


Fig. 11: Pendulum swinging in gravity: Snapshot every 0.15s.

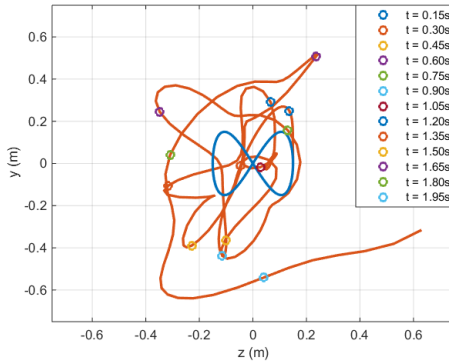


Fig. 12: Pendulum swinging in gravity: Trajectories of the two ends in the  $y - z$  plane.

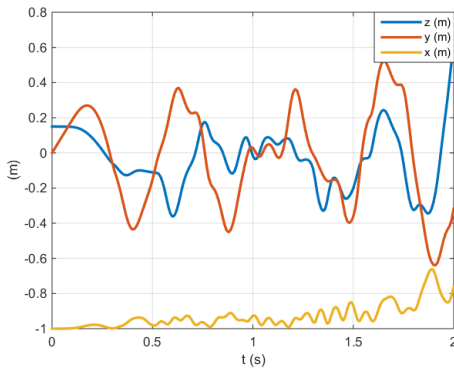


Fig. 13: Pendulum swinging in gravity: Time evolution of the tip position.

### 3) Test 6: Internally actuated floating beam in vacuum:

In this third dynamic test, one considers a free-free inextensible and non twistable Kirchhoff beam in vacuum, to which one imposes along  $K_2$  and  $K_3$ , an internal stress

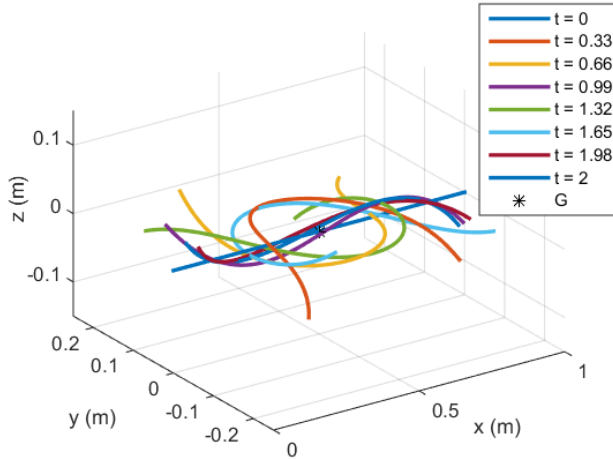


Fig. 14: Free-free internally actuated beam in vacuum: Snapshots every 0.3s in perspective view.

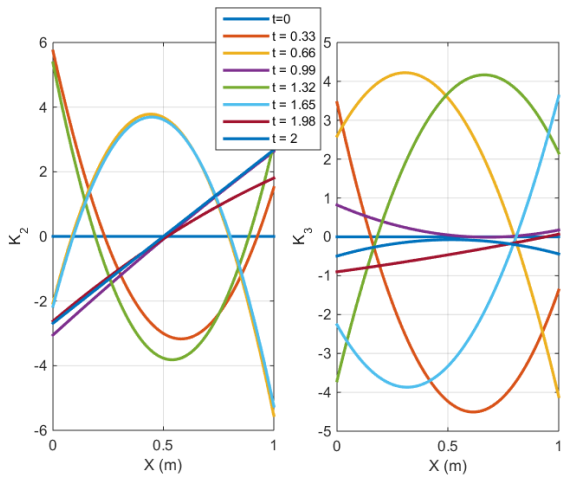


Fig. 15: Free-free internally actuated beam in vacuum:  $K_2$  (left) and  $K_3$  (right) every 0.3s.

## IX. CONCLUSION AND PERSPECTIVES

In this article we explored a new way of parameterizing an internally actuated Cosserat beam. As the PCS approach of [28], the configuration of the beam is not parameterized by the field of its cross-sectional frame poses, but rather by its strain fields. However, in contrast to the PCS, the strains are not assumed to be piecewise constant but are rather reduced on a basis of strain modes. This reduction of the original fields resembles the modal reduction of small deformations theories as the floating frame approach (FFA) of multibody systems. However, it has two main advantages over the FFA. First, the strain modes do not require to fulfill any specific boundary conditions (a constraint which is known to be one of the major drawbacks of the FFA [13]). Second, in contrast to the FFA, it is naturally suited to systems subject to deformations of high magnitudes. Based on this alternative parametrization, we propose a new

geometrically exact formulation of the statics and dynamics of an internally actuated Cosserat beam which is expected to be helpful for continuum and soft robotics. Exploiting the basic fact that the strains are the continuous counterparts of the joint angles of a discrete rigid articulated robot, this formulation has been set in the standard Lagrangian matrix form of rigid robotics. In particular, it provides a minimal set of ordinary differential equations that can be used for control design purposes, or for simulations with usual explicit time-integration schemes. Due to the relative nature of the strain-based parametrization, all the geometric nonlinearities of the Cosserat theory, are shifted from the forces related to the material (e.g. the restoring forces), to those related to space (e.g. the inertial ones), and the strain-based static formulation is much more simple than those based on the usual absolute cross sectional poses. In particular the stiffness matrix is now linear and shape independent. Thus, in statics, the approach is not only a method for constructing a Lagrange model of these systems, but also a powerful solution for their fast simulation. On the other hand, regarding dynamics, the mass matrix becomes full, nonlinearly shape dependent and difficult to compute. This issue is here tackled with a new algorithm that uses the Newton-Euler inverse dynamic model (NE-IDM) of a fully actuated beam as a computational machine for reconstructing all the matrices of its Lagrangian model. Using this algorithm for calculating inertial and external forces, the approach can be applied to bioinspired, continuum and soft robotics as well as to nonlinear structural dynamics. In this latter regard, it can be considered as an alternative to the geometrically exact method of finite elements (GE-FEM). To assess the feasibility of the approach, we numerically applied it to several benches used in the past to validate the GE-FEM in statics, and to others in dynamics or in statics, but with a beam internally actuated by pulling cables routed in its body. These first results show that with a few strain modes, one can reproduce easily the results of the GE-FEM while opening promising perspectives for statics of soft robots. In spite of these first encouraging results, this work opens several perspectives of improvement. In statics, the convergence can be improved by partitioning the beam into several sections on which the Ritz reduction would be applied. In the dynamic case, though in principle, the algorithm can be easily extended to multi-beam systems, its computational burden would practically compromise such perspectives. Several improvements could then be imagined for accelerating its computational efficiency in future. First, linearizing the NE-IDM provides a tangent algorithm that could be used (through the computation of a Jacobian) to integrate the dynamics with faster unconditionally stable implicit time-integrators. Second, while we here only considered polynomial strain basis, one could adapt such a basis to the specific systems of continuum and soft robotics (especially to their actuation), in order to reduce further the dynamic model.



In the paper a hat "∧" covering a vector  $\Upsilon$  defines a matrix  $\hat{\Upsilon} = \Upsilon^\wedge$  whose definitions depends on the dimension of  $\Upsilon$ . Namely, if  $\Upsilon \in \mathbb{R}^3$ ,  $\hat{\Upsilon}$  denotes the  $3 \times 3$  skew-symmetric matrix such that  $\hat{\Upsilon}V = \Upsilon \times V$  for any  $V \in \mathbb{R}^3$ . If  $\Upsilon = (W^T, U^T)^T \in \mathbb{R}^6$ , with  $W$  and  $U$  two any vectors of  $\mathbb{R}^3$ , then  $\hat{\Upsilon}$  is the  $4 \times 4$  matrix defined by:

$$\hat{\Upsilon} = \begin{pmatrix} \hat{W} & U \\ 0 & 0 \end{pmatrix}. \quad (66)$$

Reciprocally the superscript "∨" is such that  $\hat{\Upsilon}^\vee = \Upsilon$  for any  $\Upsilon$  in  $\mathbb{R}^3$  or  $\mathbb{R}^6$ . We also used some of the geometric notations of rigid body mechanics [35]. They essentially consist of the two operators  $ad$  and  $Ad$ , defined for any  $g = (R, r) \in SE(3)$  and  $\Upsilon = (W^T, U^T)^T$ , with  $W, U \in \mathbb{R}^3$ , by the two  $6 \times 6$  matrices:

$$Ad_g = \begin{pmatrix} R & 0 \\ \hat{r}R & R \end{pmatrix}, \quad ad_\Upsilon = \begin{pmatrix} \hat{W} & 0 \\ \hat{U} & \hat{W} \end{pmatrix}. \quad (67)$$

Since in the paper all the vectors and tensors are related to the mobile cross-sectional frames,  $ad$  is used to denote the contribution of the time ( $\Upsilon = \eta$ ) and space ( $\Upsilon = \xi$ ) derivatives of the cross sectional basis vectors, when deriving a vector with respect to the fixed frame. More occasionally, we used the finite operator  $Ad$  which allows changing a twist (velocity) vector from one frame to another, their relative pose being defined by a transformation  $g = (R, r) \in SE(3)$ . Note that  $1_{6 \times 6} + ad$  is the infinitesimal counterpart of  $Ad$ , and that for any  $s$ -parameterized curve of frames defined by the field of poses  $s \in \mathbb{R}^+ \mapsto g(s) \in SE(3)$ , we have  $dAd_g/ds = Ad_g ad_\Upsilon$  with  $\Upsilon = (g^{-1}(dg/s))^\vee$  the twist ( $s$ -rate of variation) of the frame in  $s$ , related to itself. Duality between twists and wrenches provide other useful relations with transposal matrices.

## XI. APPENDIX 2: DISCRETE *IDM* OF A FREE-FREE INTERNALLY ACTUATED COSSERAT BEAM

The forward dynamics (37) are time-integrated in state-space form with ODE45, which calls the *IDM* input-output map of the form (32). This map, which is a numerical function, is detailed below. As introduced in section 7, it consists in space-integrating in cascade 2 sets of ODEs. The first is a forward ODE. It is fed by the kinematic inputs ( $g_0 = (Q_0, r_0), \eta_0, \dot{\eta}_0, q, \dot{q}, \ddot{q}$ ) and calculates through a  $X$ -integration from 0 to 1, the kinematic fields along the beam. The second is a backward ODE initialized by the kinematics in 1. It calculates through a  $X$ -integration from 1 to 0, the field of internal wrench  $\Lambda$ , and finally the root wrench  $F_0$ ,

and the generalized internal forces  $Q_a$ , which are the outputs of the algorithm.

### ■ Forward ODE:

Inputs:  $Q_0, r_0, \eta_0, \dot{\eta}_0, q, \dot{q}, \ddot{q}$ .

$$\xi_a = \xi_{a0} + \Phi q,$$

$$\xi = B\xi_a + \bar{B}\xi_c,$$

$$(K^T, \Gamma^T)^T = \xi,$$

$$\dot{\xi} = B\Phi\dot{q},$$

$$\ddot{\xi} = B\Phi\ddot{q},$$

$$\begin{pmatrix} Q \\ r \\ \eta \\ \dot{\eta} \end{pmatrix} (0) = \begin{pmatrix} Q_0 \\ r_0 \\ \eta_0 \\ \dot{\eta}_0 \end{pmatrix}.$$

Forward integrate:

$$\frac{d}{dX} \begin{pmatrix} Q \\ r \\ \eta \\ \dot{\eta} \end{pmatrix} = \begin{pmatrix} (2\|Q\|)^{-1}A(R(Q)K)Q \\ R(Q)\Gamma \\ -ad_\xi\eta + \dot{\xi} \\ -ad_\xi\dot{\eta} - ad_\xi\eta + \ddot{\xi} \end{pmatrix},$$

Outputs:  $Q_1 = Q(1), r_1 = r(1), \eta_1 = \eta(1), \dot{\eta}_1 = \dot{\eta}(1)$ .

### ■ Backward ODE:

Inputs:  $Q_1, r_1, \eta_1, \dot{\eta}_1, q, \dot{q}, \ddot{q}$ .

$$\xi = B(\xi_{a0} + \Phi q) + \bar{B}\xi_c,$$

$$(K^T, \Gamma^T)^T = \xi,$$

$$\dot{\xi} = B\Phi\dot{q},$$

$$\ddot{\xi} = B\Phi\ddot{q},$$

$$\begin{pmatrix} Q \\ r \\ \eta \\ \dot{\eta} \\ \Lambda \\ Q_a \end{pmatrix} (1) = \begin{pmatrix} Q_1 \\ r_1 \\ \eta_1 \\ \dot{\eta}_1 \\ F_1 \\ 0 \end{pmatrix},$$

Backward integrate:

$$\frac{d}{dX} \begin{pmatrix} Q \\ r \\ \eta \\ \dot{\eta} \\ \Lambda \\ Q_a \end{pmatrix} = \begin{pmatrix} (2\|Q\|)^{-1}A(R(Q)K)Q \\ R(Q)\Gamma \\ -ad_\xi\eta + \dot{\xi} \\ -ad_\xi\dot{\eta} - ad_\xi\eta + \ddot{\xi} \\ ad_\xi^T\Lambda + \mathcal{M}\dot{\eta} - ad_\eta^T\mathcal{M}\eta - \bar{F} \\ -\Phi^T B^T \Lambda \end{pmatrix},$$

Outputs:  $(F_0, Q_a) = (-\Lambda(0), Q_a(0))$ .

## XII. ACKNOWLEDGMENT

This publication is partly supported by the Khalifa University of Science and Technology under Award No. FSU-2018-08 and Research Excellence (AARE) 2018-05.

## REFERENCES

- [1] E. Cosserat and F. Cosserat, *Théorie des corps déformables*. Hermann, Paris, 1909.
- [2] E. Reissner, "On a one-dimensional large displacement finite-strain beam theory," *Studies in Applied Mathematics*, vol. 52, no. 2, pp. 87–95, 1973.

- [3] F. Boyer and F. Renda, "Poincaré's equations for cosserat media: Application to shells," *Journal of Nonlinear Science*, 2016.
- [4] I. D. Walker, "Continuous backbone "continuum" robot manipulators," *ISRN Robotics*, vol. 2013, pp. 1–19, 2013.
- [5] R. J. Webster, J. S. Kim, N. J. Cowan, G. S. Chirikjian, and A. M. Okamura, "Nonholonomic modeling of needle steering," *Int. J. Rob. Res.*, vol. 25, no. 5-6, pp. 509–525, 2006.
- [6] M. Mahvash and P. Dupont, "Stiffness control of surgical continuum manipulators," *IEEE Trans. Robot.*, vol. 27, no. 2, pp. 334–345, Apr. 2011.
- [7] D. Rucker and R. Webster, "Statics and dynamics of continuum robots with general tendon routing and external loading," *Robotics, IEEE Transactions on*, vol. 27, no. 6, pp. 1033–1044, Dec 2011.
- [8] M. T. Chikhaoui, S. Lilge, S. Kleinschmidt, and J. Burgner-Kahrs, "Comparison of modeling approaches for a tendon actuated continuum robot with three extensible segments," *IEEE Robotics and Automation Letters*, vol. 4, no. 2, pp. 989–996, 2019.
- [9] F. Boyer, M. Porez, and W. Khalil, "Macro-continuous computed torque algorithm for a three-dimensional eel-like robot," *Robotics, IEEE Transactions on*, vol. 22, no. 4, pp. 763–775, Aug 2006.
- [10] F. Candelier, M. Porez, and F. Boyer, "Note on the swimming of an elongated body in a non-uniform flow," *Journal of Fluid Mechanics*, vol. 716, pp. 616–637, 2013.
- [11] F. Renda, M. Giorelli, M. Calisti, M. Cianchetti, and C. Laschi, "Dynamic model of a multibending soft robot arm driven by cables," *IEEE Transactions on Robotics*, vol. 30 (5), pp. 1109–1122, 2014.
- [12] D. Trivedi, A. Lot, , and C. D. Rahn, "Geometrically exact models for soft robotic manipulators," *IEEE Transactions on Robotics*, vol. 24, no. 4, pp. 773–780, 2008.
- [13] J. Simo, "A finite strain beam formulation. the three-dimensional dynamic problem. part i," *Computer Methods in Applied Mechanics and Engineering*, vol. 49, no. 1, pp. 55 – 70, 1985.
- [14] A. Cardona and M. Geradin, "A beam finite element non-linear theory with finite rotations," *International Journal for Numerical Methods in Engineering*, vol. 26, no. 11, pp. 2403–2438, 1988.
- [15] A. Ibrahimbegovic and M. Al Mikdad, "Finite rotations in dynamics of beams and implicit time-stepping schemes," *International Journal for Numerical Methods in Engineering*, vol. 41, no. 5, pp. 781–814, 1998.
- [16] F. Boyer and D. Primault, "Finite element of slender beams in finite transformations: A geometrically exact approach," *International Journal for Numerical Methods in Engineering*, vol. 59, no. 5, pp. 669–702, 2004.
- [17] V. Sonneville, A. Cardona, and O. Bruls, "Geometrically exact beam finite element formulated on the special euclidean group  $se(3)$ ," *Comput. Methods Appl. Mech. Engrg.*, vol. 268, pp. 451–474, 2014.
- [18] S. Grazioso, G. Di Gironimo, and B. Sciliano, "A geometrically exact model for soft continuum robots: The finite element deformation space formulation," *Soft Robotics*, 2018.
- [19] J. Burgess, "Bending stiffness in a simulation of undersea cable deployment," *Int. Journ. of Offshore and Polar Engineering*, vol. 3, no. 3, pp. 197–204, 1993.
- [20] Y. Sun, "Modelling and simulation of low-tension oceanic cable/body deployment," *PHD dissertation, University of Connecticut*, 1996.
- [21] C. Gatti-Bono and N. Perkins, "Physical numerical modelling of the dynamic behavior of fly line," *Journal of Sound and Vibration*, vol. 255, no. 3, pp. 555–577, 2002.
- [22] J. Till, V. Aloil, and C. Rucker, "Real-time dynamics of soft and continuum robots based on cosserat-rod models," *Inter. Journal of Robotics Research*, vol. 38, no. 6, pp. 723–746, April 2019.
- [23] M. Bergou, M. Wardetzky, S. Robinson, B. Audoly, and E. Grinspun, "Discrete elastic rods," in *ACM transactions on graphics*. ACM, 2008.
- [24] M. Gazzola, L. H. Dudte, A. G. McCormick, and L. Mahadevan, "Forward and inverse problems in the mechanics of soft filaments," *Royal Society Open Science*, vol. 5, no. 171628, 2017.
- [25] B. Siciliano and O. Khatib, *Handbook of Robotics, 2nd Ed.* Kroegeer Multimedia editor, 2016.
- [26] R. J. Webster and B. A. Jones, "Design and kinematic modeling of constant curvature continuum robots: A review," *The International Journal of Robotics Research*, vol. 29, no. 13, pp. 1661–1683, 2010.
- [27] F. Bertails, "Linear time super-helices," *Computer Graphics Forum*, vol. 28, no. 2, pp. 417–426, 2009.
- [28] F. Renda, F. Boyer, J. Dias, and L. Seneviratne, "Discrete cosserat approach for multisection soft manipulator dynamics," *IEEE Transactions on Robotics*, vol. 34, no. 6, pp. 1518–1533, Dec 2018.
- [29] F. Renda and L. Seneviratne, "A geometric and unified approach for modeling soft-rigid multi-body systems with lumped and distributed degrees of freedom," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*, May 2018, pp. 1567–1574.
- [30] M. J. Gander and G. Wanner, "From euler, ritz, and galerkin to modern computing," *SIAM review*, vol. 54, no. 4, pp. 627–666, November 16 2012.
- [31] F. Renda, M. Cianchetti, H. Abidi, J. Dias, and L. Seneviratne, "Screw-based modeling of soft manipulators with tendon and fluidic actuation," *Journal of Mechanism and Robotics*, 2017, doi:10.1115/1.4036579.
- [32] E. D. Tytell, C.-Y. Hsu, T. L. Williams, A. H. Cohen, and L. J. Fauci, "Interactions between internal forces, body stiffness, and fluid environment in a neuromechanical model of lamprey swimming," *PNAS*, vol. 107, no. 46, pp. 19832–19837, November 16 2010.
- [33] J. Canavin and P. Likins, "Floating reference frames for flexible spacecraft," *Journal of Spacecraft and Rockets*, vol. 14, no. 12, pp. 724–732, 1977.
- [34] M. W. Walker and D. E. Orin, "Efficient dynamic computer simulation of robotic mechanisms," *Journal of Dynamic Systems, Measurement, and Control*, vol. 104, no. 3, pp. 205–211, 1982.
- [35] R. M. Murray, Z. Li, and S. S. Sastry, *Robotic Manipulation*. CRC Press, 1994.
- [36] S. S. Antman, "Ordinary differential equations of nonlinear elasticity i: Foundations of the theories of non-linearly elastic rods and shell," *Arch. Rat. Mech. Anal.*, vol. 61, no. 4, pp. 307–351, 1976.
- [37] O. Ekeberg, "A combined neuronal and mechanical model of fish swimming," *Biological Cybernetics*, vol. 69, no. 5-6, pp. 363–374, 1993.
- [38] Z. D. Marchese and D. Rus, "Design, kinematics, and control of a soft spatial fluidic elastomer manipulator," *International Journal of Robotics Research*, vol. 35, no. 7, pp. 840–869, 2016.
- [39] S. D. Kelly and R. M. Murray, "Geometric phases and robotic locomotion," *Journal of Robotic Systems*, vol. 12, no. 6, pp. 417–431, 1995.
- [40] F. Renda, M. Giorelli, M. Calisti, M. Cianchetti, and C. Laschi, "Dynamic model of a multibending soft robot arm driven by cables," *Robotics, IEEE Transactions on*, vol. 30, no. 5, pp. 1109–1122, Oct 2014.
- [41] W. Khalil, "Dynamic modeling of robots using recursive newton-euler techniques," in *ICINCO2010*, 2010.
- [42] J. Y. S. Luh, M. W. Walker, and P. R. C. P. I., "On-line computational scheme for mechanical manipulator," *Trans. ASME, J. Dyn. Syst.*, vol. 102, pp. 69–76, 1980.
- [43] R. Featherstone, "The calculation of robot dynamics using articulated-body inertias," *The International Journal of Robotics Research*, vol. 2, no. 1, pp. 13–30, 1983.
- [44] F. Boyer, M. Porez, A. Leroyer, and M. Visonneau, "Fast dynamics of an eel-like robot-Comparisons with Navier-Stokes simulations," *Robotics, IEEE Transactions on*, vol. 24, no. 6, pp. 1274–1288, Dec. 2008.
- [45] J. C. Simo and L. Vu-Quoc, "A three-dimensional finite-strain rod model. Part II: Computational aspects," *Comp. Meth. Appl. Mech. Eng.*, vol. 58, pp. 79–116, 1986.
- [46] F. Boyer, M. Porez, and A. Leroyer, "Poincaré cosserat equations for the lighthill three-dimensional large amplitude elongated body theory: Application to robotics," *Journal of Nonlinear Science*, vol. 20, no. 1, pp. 47–79, 2010.
- [47] F. Candelier, F. Boyer, and A. Leroyer, "Three-dimensional extension of lighthill's large-amplitude elongated-body theory of fish locomotion," *Journal of Fluid Mechanics*, vol. 674, pp. 196–226, 2011.