Statics and Dynamics of Continuum Robots Based on Cosserat Rods and Optimal Control Theories

Frédéric Boyer, Vincent Lebastard, Fabien Candelier, Federico Renda, Mazen Alamir

Statics and dynamics of continuum robots based on Cosserat rods and optimal control theories

Frederic Boyer, Vincent Lebastard, *Member IEEE*, Fabien Candelier, Federico Renda, *Member IEEE*, Mazen Alamir.

***Abstract*—This paper explores the relationship between optimal control and Cosserat beam theory from the perspective of solving the forward and inverse dynamics (and statics as a subcase) of continuous manipulators and snake-like bio-inspired locomotors. By invoking the principle of minimum potential energy, and the Gauss principle of least constraint, it is shown that the quasi-static and dynamic evolution of these robots, are solutions of optimal control problems (OCPs) in the space variable, which can be solved at each step (of loading or time) of a simulation with the shooting method. In addition to offering an alternative viewpoint on several simulation approaches proposed in the recent past, the optimal control viewpoint allows us to improve some of them while providing a better understanding of their numerical properties. The approach and its properties are illustrated through a set of numerical examples validated against a reference simulator.**

I. Introduction

The relationships between optimal control theory (OCT) and statics of Cosserat rods are well known from the geometric control community [1]. In particular, it has been shown that the kineto-static model of a Cosserat rod consists of the Euler-Lagrange equations of an optimal control problem where the cost function is defined by its potential energy, the time by its arc length, and the optimal trajectories by the equilibrium configurations of the rod. Since the configuration space of a Cosserat rod is defined as a space of curves on $SE(3)$, the optimal control theory of left-invariant systems on Lie groups must be used to relate the two theories intrinsically [2]. As an illustration of this relationship, the search for deformed configurations of a Cosserat rod manipulated quasi-statically at both ends [3], is equivalent to that of optimal trajectories of a space vehicle on $SE(3)$ between two time-varying pauses [4]. In OCT, once the necessary first-order optimality conditions are deduced from the maximum principle, they define a Boundary Value Problem (BVP), whose solutions are the optimal trajectories that can be computed by various numerical technics, such as the shooting method [5]. Thus, once formulated as an OCP, the kineto-statics of a Cosserat rod can be solved at each step of a quasi-static simulation by applying the shooting method to its BVP. Beyond static modelling and quasi-static simulation of rods, further stability issues related to buckling [6], can be addressed with the second-order (sufficient) conditions around the optimal configurations, solutions of the first-order (necessary) ones [7].

With the emergence of continuous robotics, Cosserat rod theory has progressively imposed itself as one of the standards for modelling robots composed of rods actuated at their boundaries, or along them in a distributed way [8], [9], [10], [11], [12], [13]. In the first category, one finds the concentric tube robots (CTR) and continuum parallel robots (CPR), while the second includes tendon actuated continuum robots (TACR) [12], soft robots equipped with pressure chambers [14], or hyper-redundant swimming robots inspired of slender animals such as snakes and fish [8]. As an alternative to Lagrangian (variational) approaches such as the geometrically exact FEM [15], the discrete rod formulation [16], or the strain-based parametrization [17], the vast majority of the continuous robotics community simulates these systems by formulating their model as BVPs in the spatial variable (arc length), which are then solved at each step of the simulation by the shooting method.

OCT has been successfully applied to statics of CTRs and CPRs to study the difficult problem of their stability with second order conditions [18], [19]. However, to the best of the authors' knowledge, it has so far never been applied to the dynamics of continuous robots, while it does not seem to be mentioned in the statics of distributed actuated robots. In particular, the BVPs addressed so far in these contexts with the shooting method by the robotics community, seem not to derive from optimal control problems (OCPs), but rather to be deduced by an adhoc combination of equations based on beam kinematics and Newton's laws [20].

It is one of the main contributions of this paper, to show how, and to what extent, it is possible to model and simulate the statics and dynamics of continuous (possibly distributed actuated) robots, by deriving their BVPs systematically with the OCT, and solving them numerically with the shooting method. To illustrate our point and initiate future work, we will focus on the case of a one-piece robot which can be a manipulator (fixed at one end and free at the other) actuated by tendons, or a slender locomotor (free at both ends) actuated internally by an idealized model of the muscular activity of elongated vertebrates [21], a case never approached before with this numerical method. In statics, the

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 Univ., CNRS, IUSTI, Marseille, France. e-mail: fabien.candelier@univ-amu.fr.

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

M. Alami is with LAG-ENSIEG, CNRS, Grenoble, France. e-mail: mazen.alamir@gipsa-lab.inpg.fr.

expected extension of OCT can be achieved by modeling the distributed actuation as a stress field governed by an actuated constitutive law, as proposed in [17]. In dynamics, we will see that the forward dynamic problem of a continuous robot (manipulator or locomotor), can be reformulated as an OCP by invoking one of the founding principles of dynamics originally introduced by Gauss [22], and integrated into the framework of rational mechanics by Gibbs [23] and Appell [24]. Nowadays known as Gauss' *Principle of least constraint*, the role of this principle in the dynamics of rigid manipulators has been revealed in [25]. Remarkably, for these discrete systems, the principle leads to a linear-quadratic OCP (LQ-OCP), that once solved with the so-called sweep method [26], provides Featherstone's forward dynamic algorithm of rigid multibody systems [27]. More generally, it is possible to show that all Newton-Euler's (NE) dynamics of rigid discrete systems (from models to algorithms), are, in fact, based on Gauss principle and OCT. From this view point, a further conceptual contribution of the article consists in relating OCT and Newton-Euler dynamics in the context of continuum robotics. In particular, we will show that, as in the case of discrete rigid systems, the dynamic BVPs derived with OCT naturally contain a continuous model of the accelerations as one of the Euler-Lagrange equations, a model that is generally omitted in previous work on the shooting-based approach, and yet plays a non-negligible role in simulation.

In addition to these modeling contributions, the OC perspective will provide new insight into the numerical properties of the shooting-based approaches applied to continuous robots. In particular, we will see that the LQ-OCP underpinned by the Gauss principle applied to a continuous robot is singular, and that it can be regularized using an implicit time integration scheme. Note that a similar numerical strategy has been introduced in the ocean engineering community for towed submarine cables [28] and more recently, to solve the dynamics of continuous manipulators [20]. However, the OC point of view will show that such a regularization process is intrinsically limited by a certain critical value of the time step below which, the singularity re-expresses itself and the approach aborts, and that this critical value increases as the robot becomes increasingly soft. These numerical considerations and others are illustrated on a set of numerical benches related to fixed and floating base rods, for which an implicit geometric integration scheme on the Lie group $SO(3) \times \mathbb{R}^3$ is proposed. These benches are compared to a reference simulator based on the Lagrangian approach of [17]. Finally, the paper ends with the case of the forward dynamics of a bio-inspired swimmer, and thus completes the past results of the authors devoted to the inverse dynamics of such systems [8], while opening new perspectives for bio-inspired robotics and bio-mechanics of swimming.

The article is structured as follows. We first remind in section 2 some of the key concepts of Cosserat rods theory used in the article. In section 3, the model of distributed actuation (tendons, muscles) is also reminded and defined as a field of internal stress of a generalized active constitutive law. Based on this model, in section 4, the statics and dynamics of a single piece continuum robot are formulated in the framework of OCT, with the principle of minimum potential energy, and the Gauss least constraint principle, respectively. The BVPs provided by these different OCPs are then solved with the shooting method in section 5. While all these developments concentrate on the forward static and dynamic problems, the inverse problem is addressed in section 6 as a byproduct of the approach. In section 7, the approach is illustrated through several numerical examples.

II. Reminder of Cosserat rod theory

1) Configuration and twists: We here consider an elliptic cross-sectional[1] elastic rod subject to finite displacements and small strains[2] (see Figure 1). In the Cosserat approach, such a medium is modeled by a continuous set of rigid cross sections stacked along a material line and labelled by a coordinate $X \in [0,1]$. To each X-cross section, a mobile cross-sectional frame $\mathcal{F}(X)$ is attached. These frames are located on the center of the cross sections with their first vector normal to them and the two others aligned with the elliptic axes. In this context, the beam configuration space is naturally defined as:

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

which stands for a functional space of curves in $SE(3)$, where $g(X) = (R,r)(X)$ is represented by a 4×4 homogeneous transformation matrix (with $R(X) \in SO(3)$, and $r(X) \in \mathbb{R}^3$), parameterizing the pause of $\mathcal{F}(X)$ in $\mathcal{F}_s$. Throughout the article, we will consider continuum manipulators or locomotors. In the first case, the proximal cross section ($X = 0$) is clamped and $g(0) = 1_{4\times4}$, while in both cases, the tip cross-section ($X = 1$) is free to move. Noting $\partial./\partial X$ and $\partial./\partial t$ by a "prime" and a "dot", the field g depending on both X and t, its space-time variations are described by the two vector fields η and ξ from $[0,1]$ to $se(3) \cong \mathbb{R}^6$ (see [17] for standard Lie group notations):

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

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 $\xi = (K_1, K_2, K_3, \Gamma_1, \Gamma_2, \Gamma_3)^T = (K^T, \Gamma^T)^T$ is the exact geometrical counterpart of η when replacing t by X (see Figure 1). Since X is a material label, it is independent of t, and we have by symmetry of derivatives of transformations:

$$(g')^\cdot = (\dot{g})' \;\Rightarrow\; \dot{\xi} = \eta' + ad_\xi \eta. \tag{3}$$

Replacing t in the definition of the velocity twist η, by any real parameter ε, independent of t and X, allows defining

[1]This choice will allow to model continuous swimmers inspired from fish as addressed in one of the numerical examples of section VII.

[2]This assumption which is systematically made in the Cosserat rod theory typically means that the radius of curvature along the rod is of the order of its length.

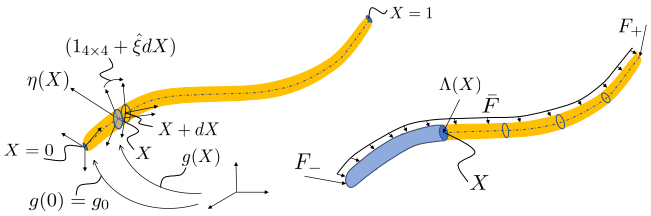


Fig. 1: Twists (left) and wrenches (right) of a Cosserat rod.

what we name a variation $\delta g = g\delta\hat{\zeta}$ of the rod configuration. Such a δg generates on a functional f of g, a variation δf defined for all X by:

$$\delta f(g(X)) = \frac{d}{d\varepsilon}\bigg|_{\varepsilon=0} f\left(g(X)\exp(\varepsilon\delta\hat{\zeta}(X))\right). \quad (4)$$

The parameter ε of a δ-variation being independent of X, a similar commutation relation to (3) holds where the time-derivation $\partial/\partial t$, is simply replaced by the variation δ:

$$\delta(g') = (\delta g)' \;\Rightarrow\; \delta\xi = \delta\zeta' + ad_\xi\delta\zeta. \quad (5)$$

Relations (3) and (5) are due to Poincaré [29] and play a key role in the modern Euler-Poincaré Lagrangian reduction theory [30]. In this context, they allow applying Hamilton's variational principle to Cosserat rods directly on the configuration space (1), where $\delta\zeta = (g^{-1}\delta g)^\vee$ are then, some virtual displacements of the rod configuration [31]. In section IV.A, devoted to statics, we adopt the view point of optimal control theory. In this context, we will also use configuration variations $\delta\zeta$, for which (5) holds. However, in contrast to Lagrangian mechanics, these variations are not directly applied to the configuration of the rod as this is the case of virtual displacements, but are the indirect consequences of some variations of its strain field.

2) Strains and stress: The Cosserat rod model can capture several rod sub-models (e.g. Kirchhoff) by allowing only a subset of the components of ξ to be free. Referring to [17], these allowed components are gathered in a $n_a \times 1$ ($n_a \leq 6$) vector field noted ξ_a, while the other components (that are constrained to some constant values), defines a complementary $n_c \times 1$ field, noted ξ_c, with $n_c = 6 - n_a$. With these definitions in hands, we have the partition of ξ:

$$\xi = B\xi_a + \bar{B}\xi_c, \quad (6)$$

where B and $\bar{B}$ are two selection matrices of dimension $6 \times n_a$ and $6 \times n_c$, with entries equal to zero or one, while ξ_c is a constant vector field whose components (zero or constant), depend on the adopted rod kinematics (e.g. for a Kirchhoff rod, $B = (1_{3\times3}, 0_{3\times3})^T$, $\bar{B} = (0_{3\times3}, 1_{3\times3})^T$, and if X is the arc length along the rod $\xi_c = (1,0,0)^T$). If g_o defines a reference stress-less configuration of the rod with space-twist field $\xi_o = (g_o^{-1}g_o')^\vee$, a linear strain measure is defined by the field:

$$\epsilon = \xi_a - \xi_{a,o} = B^T(\xi - \xi_o). \quad (7)$$

Combining (6) and (7), one can express the field ξ and its time derivatives, in terms of strains as:

$$\xi = B(\epsilon + \xi_{a,o}) + \bar{B}\xi_c = B\epsilon + \xi_o,$$
$$\dot{\xi} = B\dot{\xi}_a = B\dot{\epsilon}\ ,\ \ddot{\xi} = B\ddot{\xi}_a = B\ddot{\epsilon}. \quad (8)$$

The stress state is described by a field of wrench $\Lambda : X \in [0,1] \mapsto \Lambda(X) = (C^T, N^T)^T(X)$, where $C = (C_1, C_2, C_3)^T$, and $N = (N_1, N_2, N_3)^T$, are the moment and the resultant of the internal tension forces transmitted from right ($Y > X$), to left ($Y < X$), across the X-cross section, and expressed in its mobile frame (see Figure 1). According to (6), one can partition Λ as:

$$\Lambda = B\Lambda_a + \bar{B}\Lambda_c, \quad (9)$$

where $\Lambda_a = B^T\Lambda$, gathers the components of Λ acting along the allowed internal d.o.f of ξ_a, while Λ_c defines a set of Lagrange multipliers (internal forces and/or reaction torques) in charge of forcing the internal stresses $\bar{B}^T\xi = \xi_c$.

3) Forces and energies: In statics, we only consider the case of a manipulator subject to a density of wrench $\bar{F}$ and a tip wrench F_+ applied along its length, and at its distal end respectively. In all the subsequent developments, we assume $\bar{F}$ and F_+ to be prescribed by some explicit functions of time and/or defined by a state-dependent model, where the state is g in statics, and (g, η) in dynamics. In statics, a force $F(g)$ exerted on a single cross section of pause g, is said to be conservative if it derives from a potential, i.e. if there exists a function U_{ext} of $SE(3)$ in $\mathbb{R}$, such that:

$$\delta\zeta^T F(g) = -\delta U_{\text{ext}}, \quad (10)$$

where δU_{ext} is defined by instantiating f by U_{ext} in (4). In the following developments on statics, F_+ and $\bar{F}$ are assumed to be conservative, i.e. it exists U_{ext} for the entire rod, such that:

$$\delta U_{\text{ext}} = \int_0^1 \delta\bar{\mathfrak{U}}\, dX + \delta U_+, \quad (11)$$

with:

$$\delta\bar{\mathfrak{U}} = -\delta\zeta^T\bar{F}\ \ ,\ \delta U_+ = -\delta\zeta(1)^T F_+. \quad (12)$$

Assuming the rod to be elastic, and subject to small strains, its internal (potential) energy is defined as the integral over the rod of a quadratic form of the strains:

$$U_{\text{int}} = \int_0^1 \mathfrak{U}_{\text{int}}\, dX = \frac{1}{2}\int_0^1 \epsilon^T\mathcal{H}_r\epsilon\, dX, \quad (13)$$

where $\mathcal{H}_r = B^T\text{diag}(GJ_1, EJ_2, EJ_3, EA, GA, GA)B$ is the reduced $n_a \times n_a$ Hooke matrix of the rod. Moreover, in this case, the field of stress Λ_a is also conservative, and so derives from the potential (13), i.e. we have:

$$\Lambda_a = B^T\Lambda = \frac{\partial\mathfrak{U}_{\text{int}}}{\partial\epsilon} = \mathcal{H}_r\epsilon. \quad (14)$$

Similarly, the kinetic energy of the rod is:

$$T = \frac{1}{2}\int_0^1 \eta^T\mathcal{M}\eta\, dX, \quad (15)$$

where $\mathcal{M} = \text{diag}(\rho J_1, \rho J_2, \rho J_3, \rho A, \rho A, \rho A) = \text{diag}(\rho J, \rho A 1_{3\times3})$, is the field of 6×6 inertia matrix of its cross-sections.

III. Continuum robots as internally actuated Cosserat rods

We consider a continuum robot as an elastic slender body equipped with some active devices able to exert some forces on it. The body is modelled as a Cosserat rod, actuated with "external" or "internal" actuators, depending whether the forces they exert on the robot, are supported by the external environment, or by the constitutive material of the rod itself. As emblematic examples of each category, rods equipped with magnetic dipoles in interaction with a controlled external magnetic field are externally actuated. On the other hand, a continuum manipulator consisting of a rod made with a piezoelectric material controlled in voltage is internally actuated. Beyond robotics, the muscles contractions that bend the slender body of a snake or an eel, define an internal actuation. To derive the model of a continuum robot, it suffices to modify the model of (passive) Cosserat rods in two different ways, depending on the nature of actuation. If it is external, we add to the external wrenches $\bar{F}$ (gravity, contact..), a field of external actuation wrench $\bar{F}_d$. If it is internal, we add to the field of elastic stress $B^T\Lambda = \Lambda_a = \mathcal{H}_r\epsilon$, a further exogenous active component Λ_{ad} modelling the actuation. In this case, the passive constitutive law (14) is changed into the active/passive one [17]:

$$B^T\Lambda = \Lambda_{ad} + \mathcal{H}_r\epsilon. \tag{16}$$

Physically, when the rod kinematics are not constrained (i.e. $B = 1_{6\times 6}$), (16) simply means that if we (virtually) cut a slender continuous robot into two parts at any point X along its dominant length, the part of the robot $Y > X$ exerts on the part $Y < X$ a wrench defined by (16) evaluated in X, i.e., with two components, one ($\mathcal{H}_r\epsilon$) modeling the elastic restoring forces and torques of the structure, the other (Λ_{ad}) modeling the effects of any internal actuating device transmitting forces along it. Remarkably, some continuous manipulators such as TACRs can be classified in both categories, i.e. the actuation forces can be modeled either by modifying $\bar{F}$ or by using a constitutive law of the form (16). To illustrate this context, let us consider a TACR shape controlled by pulling a set of N tendons of negligible inertia, elasticity and friction. In the first view point (external), Newton's law are first applied to the tendons and the rod alone, and action-reaction is then used to get the field of external wrench exerted by each tendon onto the rod [12]. In the second (internal), the rod and tendons are considered as a single system to which Newton [32], or Lagrangian mechanics is applied [17]. In this later context, invoking kinematic invariance of virtual works between the space of tendon lengths and that of Cosserat strains, provides the expression of the internal stress-wrench field Λ_{ad} as a function of the tensions T_i along the tendons (the subscript "c" here means "cable"):

$$\Lambda_{ad}(X,t) = \sum_{i=1}^{N} B^T \begin{pmatrix} D_i \times \Gamma_{c,i} \\ \Gamma_{c,i} \end{pmatrix} \frac{T_i(t)}{\|\Gamma_{c,i}\|}, \tag{17}$$

where we introduced the notation $\Gamma_{c,i} = \Gamma + K \times D_i + D_i'$[4], with $D_i(X)$ the position of the intersection point of cable i with the X-cross section of the rod, in the X-cross-sectional frame. Note that although Λ_{ad} depends on ϵ due to the presence of K in $\Gamma_{c,i}$, more detailed computations show that this dependence vanishes as soon as the routings of the cables are parallel with the rod backbone, while due to small radius of TACRs, they are very small otherwise. In the following, the case of a TACR in which the ϵ-dependency of Λ_{ad} is neglected, is used as an emblematic example of the approach. However, beyond this archetypal example, all the subsequent results can be applied to any continuous slender robot whose internal stress (elastic and actuated) can be modeled by a nonlinear version $B^T\Lambda = f(\epsilon, \tau(t))$ of (16), with $\tau(t)$ a finite vector of control inputs (e.g., tendon tensions, chamber pressures...). The key to extending the approach to this broader context will be provided in a future remark numbered 4, the goal of which is to capture the ϵ dependence of (17) throughout the approach[3]. In summary, apart from remark 4, a continuum robot will in the following be considered as a Cosserat rod internally actuated by (16), where Λ_{ad} is a dependent function of (X,t) modeling a technological action, or an idealization of the muscular activity of a slender animal.

IV. Forward statics and dynamics of continuum robots as OCPs

We now address the forward statics and dynamics of continuum robots as some optimal control problems (OCP). For obvious reasons, the case of a locomotor is only considered in dynamics. In statics, the cost functional is the potential energy of the rod, in dynamics, it is the Gauss constraint.

A. Statics of a continuum robot

In statics, a material system subject to conservative forces is governed by the principle of minimum potential energy, which states that among all the configurations accessible to the system, the equilibrium configuration it occupies must make its potential energy stationary. In the case of a continuum manipulator subject to external and internal forces governed by (12) and (16) respectively, its potential energy takes the form:

$$U = U_{\text{int}} + U_{\text{ext}} = \int_0^1 \frac{1}{2}\epsilon^T\mathcal{H}_r\epsilon + \epsilon^T\Lambda_{ad} + \bar{\mathfrak{U}}\, dX + U_+. \tag{18}$$

Now, giving to the strain ϵ the role of a control variable u, and to g, that of a state variable, the principle of minimum potential energy can be reformulated as the following OCP, where the space variable X replaces the usual time in control:

[3]Note that this context is also that of a certain class of fluid-actuated continuous robots where tendons are replaced by radially incompressible pressure-controlled tubular chambers [32].

• ***OCP1****: Find the optimal control u that makes stationary the cost functional defined by the potential energy:*

$$\mathcal{C}(u) = \int_0^1 \frac{1}{2} u^T \mathcal{H}_r u + u^T \Lambda_{ad} + \bar{\mathfrak{U}} \, dX + U_+ \tag{19}$$

under the constraints:

$$\begin{aligned} g' = g(Bu + \xi_o)^\wedge \;\Leftrightarrow\; (g^{-1}g')^\vee = (Bu + \xi_o) \\ \Leftrightarrow\; \xi = Bu + \xi_o \;, \end{aligned} \tag{20}$$

with $g(0) = 1_{4\times4}$, due to the fixed root cross-section •

While (19) defines the potential energy of the continuum manipulator, (20) simply describes how the pause cross sections evolve with X. In particular, (20) accounts for the internal kinematic constraints imposed by the choice of the selection matrix B, as its inextensibility or unshearibility. Since the control is unbounded, the necessary conditions that any optimal u needs to fulfill can be derived by applying usual variational calculus on the Lie group $SE(3)$, to the augmented cost functional:

$$\mathcal{C}_+ = \int_0^1 \frac{1}{2} u^T \mathcal{H}_r u + u^T \Lambda_{ad} + \bar{\mathfrak{U}} + (\xi - \xi_o - Bu)^T \Lambda \; dX + U_+, \tag{21}$$

with Λ a field of Lagrange multipliers. This approach avoids the (Hamiltonian) Lie-Poisson reduction, in favor of the (Lagrangian) Euler-Poincaré one [33]. In this setting, the optimal control must satisfy the stationarity condition of the functional $\mathcal{C}_+$:

$$\begin{aligned} \delta\mathcal{C}_+ = \textstyle\int_0^1 \delta u^T (\Lambda_{ad} + \mathcal{H}_r u)\, dX + \\ + \textstyle\int_0^1 \delta\bar{\mathfrak{U}} + (\delta\xi - B\delta u)^T \Lambda \; dX + \delta U_+ = 0, \end{aligned} \tag{22}$$

where for any function $f(g)$, variations δf are consequences of variations δu in (20). Now, since these variations do not affect X, they generate some pose variations $\delta\zeta = (g^{-1}\delta g)^\vee$ which satisfy (5), and (22) can be rewritten:

$$\begin{aligned} \delta\mathcal{C}_+ = \textstyle\int_0^1 \delta u^T (\Lambda_{ad} + \mathcal{H}_r u) + \delta\bar{\mathfrak{U}}\, dX + \\ \textstyle\int_0^1 (\delta\zeta'^T + \delta\zeta^T ad_\xi^T - B\delta u)^T \Lambda \, dX + \delta U_+ = 0, \end{aligned} \tag{23}$$

which gives, after by-part integration and with the definition of conservative external forces (12):

$$\begin{aligned} 0 = \textstyle\int_0^1 \delta u^T (\Lambda_{ad} + \mathcal{H}_r u - B^T \Lambda) dX \\ - \textstyle\int_0^1 \delta\zeta^T (\Lambda' - ad_\xi^T \Lambda + \bar{F}) dX + \delta\zeta^T(1)(\Lambda(1) - F_+), \end{aligned} \tag{24}$$

where we used the fact that the BC at $X = 0$, imposes $\delta\zeta(0) = (g^{-1}\delta g)^\vee(0) = 0$. It would be tedious to determine the variation $\delta\zeta$ produced by a given δu, so we choose Λ to cause the coefficients of $\delta\zeta$ in (24) to vanish [26]. After this choice, (24) only imposes the coefficient of δu to be zero, and the conditions of stationarity that any optimal trajectory must satisfy are (remind that $\xi = (g^{-1}g')^\vee$):

• The two Euler-Lagrange equations :

$$\begin{pmatrix} g' \\ \Lambda' \end{pmatrix} = \begin{pmatrix} g(\xi_o + Bu)^\wedge \\ ad_\xi^T \Lambda - \bar{F} \end{pmatrix}, \tag{25}$$

• The optimality condition:

$$B^T \Lambda = \Lambda_{ad} + \mathcal{H}_r u, \tag{26}$$

• The transversality condition:

$$\Lambda(1) = F_+. \tag{27}$$

Once supplemented with the geometric BC: $g(0) = 1_{4\times4}$, this set of equations defines a closed formulation of the statics of a continuum robot. Finally, solving this formulation, or solving OCP1, is equivalent. Achieving such a resolution, numerically at each loading step, i.e. when the internal actuation stress, the density of external wrench and the external wrench at the tip Λ_{ad}, $\bar{F}$ and F_+ are updated, provides a simulation algorithm of the quasi-static evolution of a continuum manipulator.

Remark 1: In the passive case of Kirchhoff rods straight at rest, i.e. when $\Lambda_{ad} = 0$, $B = (1_{3\times3}, 0_{3\times3})^T$, and $\xi_0 = (0_{1\times3}, 1, 0, 0)^T$, this formulation has been derived from optimal control theory of left-invariant systems on Lie groups in [2], and applied in robotics [3] with a second geometric BC instead of (27), in order to address the difficult problem of controlling the shape of a rod manipulated at its two ends [34]. In continuum robotics it has been exploited in [18] and [19], in order to analyse the stability of concentric tubes robots (CTR), and continuum parallel robots (CPR) respectively, i.e. continuum robots constituted of several passive rods actuated by exerting localized torques and forces on their boundaries. In these two cases, the above first order optimality conditions are augmented of second order Legendre-Clebsh conditions, which once supplemented with a study of conjugate points (the so-called Jacobi conditions), allow to analyse the stability of these systems [35]. In the case of systems actuated in a distributed way, as TACRs, a similar BVP is considered for static simulation, but with actuation modelled as external forces (see section III), and with no reference to OC [12].

B. Dynamics of a continuum robot

To extend the previous picture of statics to dynamics, we must introduce a little known variational principle proposed by Gauss in 1829 and known today as the "principle of least constraint". In words this principle can be stated as follows: *Let us consider a constrained material system that has reached at a given time t its current state, then, among all the accelerations compatible with the constraints that the system can have at t, that it will have is the closest to that it would have if all the constraints would be instantaneously removed at that time* [36]. Remarkably, this principle can ground dynamics as does Hamilton's principle in Lagrangian mechanics. However, in contrast to Hamilton's principle, the Gauss principle is not an extremal integral principle, but rather a minimal differential principle, that can be formally stated at any instant of the motion of a system subject to constraints (holonomic or not), in the form:

$$a = \arg\min_{a_c} \left(\frac{1}{2} \|a_c - a_f\|_K^2 \right), \tag{28}$$

where a_c, $a_f \in \mathbb{R}^3$ stand for the acceleration field of the system, with, and without constraints respectively, while a stands for its actual acceleration field, and $\|.\|_K$ is the norm defined by its kinetic energy. The minimized functional is named the "Gauss constraint". Going further into details, consider a material system of volume element dv with mass density ρ, applying (28) to this system leads to:

$$\begin{aligned} a &= \arg\min_{a_c} \left(\tfrac{1}{2}\int_{\mathcal{D}}(a_c - a_f)^2 \rho\, dv\right) \\ &= \arg\min_{a_c} \left(\int_{\mathcal{D}}(\tfrac{1}{2}\rho\, a_c^2 - a_c^T \bar{f})dv\right), \end{aligned} \tag{29}$$

where $\bar{f} = \rho\, a_f$ denotes the volume density of all external and internal forces, except for the "reactive forces" produced by the constraints[4], while the term a_f^2 can be ignored since it does not depend on the optimization variable a_c. In this equivalent formulation, the first integral term is named the "acceleration energy" by Gibbs and Appell [23], [24], and here noted T_{acc}. By extension, the opposite of the second term of (29) could be named the "acceleration power of external and internal non-reactive forces" and noted P_{acc}. By "non-reactive" forces we here mean those which do not contribute to force the constraints. Practically, T_{acc} and P_{acc} can be easily computed by replacing first, the velocities by accelerations in the usual definitions of kinetic energy and (non-reactive) power, and second, by removing the terms that only depend on velocities, in the final expressions[5]. Applying these computational rules to a continuum robot, these definitions become (see Appendix 1):

$$T_{\text{acc}} = \frac{1}{2}\int_{\mathcal{D}} \rho\, a_c^2 dv = \int_0^1 \frac{1}{2}\dot{\eta}^T \mathcal{M}\dot{\eta} - \dot{\eta}^T (ad_\eta^T \mathcal{M}\eta) dX, \tag{30}$$

and:

$$\begin{aligned} P_{\text{acc}} = \textstyle\int_{\mathcal{D}} a_c^T \bar{f} dv = \\ \textstyle\int_0^1 \dot{\eta}^T \bar{F} + \ddot{\xi}_a^T (\Lambda_{ad} - \mathcal{H}_r \epsilon)\; dX + \dot{\eta}(1)^T F_+. \end{aligned} \tag{31}$$

if the robot is a manipulator, or:

$$\begin{aligned} P_{\text{acc}} = \textstyle\int_{\mathcal{D}} a_c^T \bar{f} dv = \int_0^1 \dot{\eta}^T \bar{F} + \ddot{\xi}_a^T (\Lambda_{ad} - \mathcal{H}_r \epsilon)\; dX + \\ + \dot{\eta}(0)^T F_- + \dot{\eta}(1)^T F_+, \end{aligned} \tag{32}$$

if it is a locomotor. Note here, that the constraints being those imposed by the geometric BCs and the internal rod kinematics (e.g., inextensibility and unshearability of a Kirchhoff rod), the reactive forces they produce, do not appear in P_{acc}. Applying the Gauss principle to the forward dynamics of continuum robot, leads to find at each time t of the robot motion, the strain acceleration $\ddot{\epsilon}$ that minimizes the functional:

$$\frac{1}{2}\int_{\mathcal{D}} (a_c - a_f)^2 \rho\, dv = T_{\text{acc}} - P_{\text{acc}}, \tag{33}$$

with T_{acc} and P_{acc} defined by (30) and (31,32), and where a_c being compatible with Cosserat kinematics (see Appendix 1), $\dot{\eta}$ needs to fulfill the constraints on accelerations deduced by time-differentiating (3):

$$\dot{\eta}' = -ad_\xi \dot{\eta} - ad_{\dot{\xi}} \eta + B\ddot{\epsilon} \quad , \quad \dot{\eta}(0) = 0, \tag{34}$$

where $\ddot{\epsilon}$ plays the role of a minimal set of accelerations parameterizing any acceleration field compatible with the internal constraints imposed by the rod model. Moreover, interpreting this field of acceleration strain as a control input $u = \ddot{\xi}_a = \ddot{\epsilon}$, the principle turns to be an optimal control problem that can be stated as follows.

• ***OCP2****: Find at each time t of the robot motion, the strain acceleration u that minimizes:*

$$\begin{aligned} \mathcal{C}(u) = \textstyle\int_0^1 \frac{1}{2}\dot{\eta}^T \mathcal{M}\dot{\eta} - \dot{\eta}^T (\bar{F} + ad_\eta^T \mathcal{M}\eta) dX \\ + \textstyle\int_0^1 u^T (\Lambda_{ad} - \mathcal{H}_r \epsilon) dX - \dot{\eta}(1)^T F_+, \end{aligned} \tag{35}$$

if the robot is a manipulator, or:

$$\begin{aligned} \mathcal{C}(u) = \textstyle\int_0^1 \frac{1}{2}\dot{\eta}^T \mathcal{M}\dot{\eta} - \dot{\eta}^T (\bar{F} + ad_\eta^T \mathcal{M}\eta) dX \\ + \textstyle\int_0^1 u^T (\Lambda_{ad} - \mathcal{H}_r \epsilon) dX - \dot{\eta}(0)^T F_- - \dot{\eta}(1)^T F_+, \end{aligned} \tag{36}$$

if it is a locomotor. In all cases (manip. or locom.), this has to be done under the constraint:

$$\dot{\eta}' = -ad_\xi \dot{\eta} - ad_{\dot{\xi}} \eta + Bu \;, \tag{37}$$

and the further one:

$$\dot{\eta}(0) = 0, \tag{38}$$

if the robot is a manipulator •

Note that in this formulation, the mechanical state (g, η) of the robot is frozen at the current time t, which means that all the (g, η)-dependent functions of (35,36,37) (including $\epsilon = B^T (g^{-1} g')^\vee - \xi_{ao}$), can be considered as some X-dependent functions. As a result $\dot{\eta}$ stands for the state vector of an optimal linear-quadratic (LQ)-OCP, in which, like in the static case, X plays the role of time. Using usual control notations, the linear state system[6] is defined by (37) and takes the form:

$$\dot{x} = A(t)x + B(t)u + c(t), \tag{39}$$

with the correspondence between notations as indicated in table 1, and where the state-independent drift vector c, can be easily removed by a change of variable on the accelerations $\dot{\eta}$ as done in the discrete case in [25]. As regards the quadratic cost (35) or (36), it is of the generic Bolza form [26]:

$$\begin{aligned} \mathcal{C}(u) = \textstyle\int_0^1 \frac{1}{2} x^T M(t) x\, dt \\ - \textstyle\int_0^1 x^T h(t) + u^T f(t)\, dt + \phi(x(0), x(1)), \end{aligned} \tag{40}$$

where we used the notations of table 1. The control variable u being unbounded, one can still apply direct variational calculus to obtain the first order necessary conditions that any

[4] In the Lagrangian terminology, these reactive forces are the Lagrange multipliers in charge of forcing the constraints.

[5] These terms play no role in Gauss principle since they add meaningless constants to the Gauss constraint.

[6] Referring to OC, where the time replaces X, this is a "time"-variant linear system.

OCP2	Gauss principle
u	$\ddot{\xi}_a = \ddot{\epsilon}$
t	X
h	$\bar{F} + ad_\eta^T \mathcal{M}\eta$
x	$\dot{\eta}$
f	$\Lambda_{ad} - \mathcal{H}_r\epsilon$
M	$\mathcal{M}$
A	$-ad_\xi$
B	$\bar{B}$
ϕ (manip.)	$-\dot{\eta}(1)^T F_+$
ϕ (locom.)	$-\dot{\eta}(0)^T F_- - \dot{\eta}(1)^T F_+$

TABLE I: Table of correspondences between the linear optimal control problem OCP2 and the Gauss "least constraint" principle.

optimal control u needs to fulfill [37]. These conditions can be expressed with the (control) Hamiltonian of a continuum robot:

$$H(\dot{\eta}, \Lambda, u) = L(\dot{\eta}, u) + \Lambda^T(-ad_\xi\dot{\eta} - ad_{\dot{\xi}}\eta + Bu), \quad (41)$$

with L, the Lagrangian of the problem defined as the function under the integral of (35) or (36), and where, as in the static case, the stress field Λ defines the costate of the optimal problem. Then, remarking that the state $\dot{\eta}$ now belongs to a vector space, usual variational calculus provides the classical first order conditions [26]:

$$\dot{\eta}' = \frac{\partial H}{\partial \Lambda} \ , \ \Lambda' = -\frac{\partial H}{\partial \dot{\eta}} \ , \ \frac{\partial H}{\partial u} = 0, \quad (42)$$

as well as the transversality condition:

$$\Lambda(1) = \frac{\partial \phi}{\partial \dot{\eta}(1)}, \quad (43)$$

for a manipulator, or:

$$\Lambda(0) = -\frac{\partial \phi}{\partial \dot{\eta}(0)} \ , \ \Lambda(1) = \frac{\partial \phi}{\partial \dot{\eta}(1)}, \quad (44)$$

for a locomotor. From left to right, the two first equations of (42) are two Euler-Lagrange ODEs governing the state $\dot{\eta}$ and costate Λ vectors, while the third stands for an algebraic condition that an optimal control u must hold in any point of an optimal $(\dot{\eta}, \Lambda)$ trajectory. Introducing (41) into these formulas, and changing the orientation of the stress, (i.e. Λ into $-\Lambda$ and Λ_{ad} into $-\Lambda_{ad}$), which is a matter of convention[7], provides:

• The two Euler-Lagrange equations:

$$\begin{pmatrix} \dot{\eta}' \\ \Lambda' \end{pmatrix} = \begin{pmatrix} -ad_\xi & 0 \\ \mathcal{M} & ad_\xi^T \end{pmatrix} \begin{pmatrix} \dot{\eta} \\ \Lambda \end{pmatrix} + \begin{pmatrix} -ad_{\dot{\xi}}\eta + Bu \\ -ad_\eta^T \mathcal{M}\eta - \bar{F} \end{pmatrix}. \quad (45)$$

• The optimality condition:

$$B^T\Lambda = \Lambda_{ad} + \mathcal{H}_r\epsilon. \quad (46)$$

[7]Conventionally, in Cosserat theory, $\Lambda(X)$ represents the wrench of the internal contact forces exerted by the piece of rod $Y > X$ onto the piece $Y < X$, across the X-cross section, while in Gauss principle this is the opposite convention which holds.

• The transversality and geometric boundary conditions: [7]

$$\Lambda(1) = F_+ \ , \ g(0) = 1_{4\times4}. \quad (47)$$

for a manipulator, or:

$$\Lambda(0) = -F_- \ , \ \Lambda(1) = F_+. \quad (48)$$

for a locomotor. Solving this formulation is equivalent to solve OCP2, i.e. an optimal control problem, here based on Gauss principle.

Remark 2: The above formulation (45-48) holds at any fixed time. To use it continuously along a time interval, we need to add a further set of equations. To introduce this point, let us consider the problem of the dynamic simulation of a manipulator. If at a given time t, one has solved (45-47). Then $u(t) = \ddot{\epsilon}(t) = \ddot{\xi}_a(t)$ is known and can be time-integrated twice with an explicit scheme, to give $(\xi_a, \dot{\xi}_a)(t + \Delta t)$. Then, to update (45-47) at $t + \Delta t$ (and resume the process), we need to reconstruct $(g, \eta)(t + \Delta t)$ from the knowledge of $(\xi_a, \dot{\xi}_a)(t+\Delta t)$. This is achieved by X-integrating from $X = 0$ to 1, the continuum kinematic models of transformations and velocities deduced from (2) and (3) respectively, and initialised with $(g, \eta)(0) = (1_{4\times4}, 0)$:

$$\begin{pmatrix} g' \\ \eta' \end{pmatrix} = \begin{pmatrix} g\hat{\xi} \\ -ad_\xi\eta + \dot{\xi} \end{pmatrix}, \quad (49)$$

where remind that $\xi = B\xi_a + \bar{B}\xi_c$ and $\dot{\xi} = B\dot{\xi}_a$. As a result, to be used in a simulation time-loop, the above set of equations (45-48) needs to be supplemented with the reconstruction equations (49). This is in contrast to OCP1 where the Euler-Lagrange equations contained all the information required by a (static) simulation, and can be explained by the fact that, while in statics the mechanical state and the control state are identical (and coincide with g), in dynamics, the control state is defined by the pose acceleration $\dot{\eta}$, while the mechanical state is (g, η).

Remark 3: In bio-robotics, the formulation (45-49) with the reduced stiffness matrix $\mathcal{H}_r = 0$, has been used as a continuous Newton-Euler (NE) model, to inverse the dynamics of hyper-redundant locomotors inspired from fish [8] and snakes [38]. As for discrete rigid multibody systems (MBS), such a model consists of the NE equations of the rigid bodies (here the cross sections), and of recursions (here ODEs) on their pauses, velocities and accelerations[8]. In Section VI, a new solution to the inverse dynamics of continuous robots will be given. In [20], a slightly different formulation is extensively used for the simulation of continuum robots. This formulation differs from the one derived above in that the model of $\dot{\eta}$ accelerations (the upper ODE of (45)) is ignored, while distributed actuation (e.g., by tendons) is not modeled by Λ_{ad}, but rather by $\bar{F}$, as discussed in Section III. Inherited from the ocean engineering community

[8]One can deduce this continuous NE model from that of a rigid multibody system, by taking an infinite number of infinitely small bodies.

interested in submarine cables [39], this alternative dynamic formulation does not derive from an OCP.

OCP	Statics: OCP1	Dynamics: OCP2
Cost funct.	Potential energy (19)	Gauss constraint (35) or (36)
State	$g \in SE(3)$	$\dot{\eta} \in se(3) \cong \mathbb{R}^6$
Costate	$\Lambda \in se(3)^\star \cong \mathbb{R}^6$	$\Lambda \in se(3)^\star \cong \mathbb{R}^6$
State-eq.	(25.top)	(45.top)
Costate-eq.	(25.bottom)	(45.bottom)
Optim.-cond.	(26)	(46)
BVP	(25-27)	(45-48)

TABLE II: Table of correspondences between the static and dynamic OCPs.

Before addressing the resolution of OCP1 and OCP2, the role of mathematical relations and objects from the point of view of optimal control, are listed in statics and dynamics in Table 2, which thus allows the two contexts to be compared.

V. Resolution of the OCPs for simulation of continuum robots

Using indirect methods for solving OCPs [5], the resolution progresses in two stages. The first consists in using the optimality condition to remove u from the state and costate equations. Once this is achieved, these two equations define an autonomous BVP, which can be solved by different numerical technics (collocation, shooting method, finite differences...). This numerical resolution defines the second stage of the resolution of the OCP. In fine, the optimality condition can be reused in an inverse way in order to compute u. We are now going to see how this resolution method can be applied to static and dynamic simulation of continuum robots. In section V.A and V.B we first apply the first stage of the approach in order to produce some autonomous BVPs for OCP1 and OCP2, whose numerical resolution is addressed with shooting method in section V.C.

A. Autonomous BVP for OCP1

In the static case, the above approach can be applied with no difficulty. Indeed, using the condition (26) provides the elimination relation:

$$B^T\Lambda = \Lambda_{ad} + \mathcal{H}_r u \;\Rightarrow\; u = \mathcal{H}_r^{-1}(B^T\Lambda - \Lambda_{ad}), \quad (50)$$

which once introduced in (25) provides the autonomous BVP:

$$\begin{pmatrix} g' \\ \Lambda' \end{pmatrix} = \begin{pmatrix} g(B\mathcal{H}_r^{-1}(B^T\Lambda - \Lambda_{ad}) + \xi_o)^\wedge \\ ad_\xi^T\Lambda - \bar{F} \end{pmatrix}, \quad (51)$$

with BCs:

$$g(0) = 1_{4\times 4}\ ,\ \Lambda(1) = F_+. \quad (52)$$

Finally, to achieve the resolution of OCP1 governing statics of a manipulator, this nonlinear BVP needs to be numerically solved, e.g. with the shooting method as detailed in the section V.C.

Remark 4: When Λ_{ad} depends on $u = \epsilon$, the optimality condition (50) becomes an implicit relation and one cannot directly use it to eliminate u in the BVP. In such a case, one could use a further implicit solver, or more simply, reexpress the BVP (25-27) in terms of strains ϵ, instead of stress Λ, by directly introducing the full constitutive law (9,16):

$$\Lambda = B(\Lambda_{ad} + \mathcal{H}_r\epsilon) + \bar{B}\Lambda_c, \quad (53)$$

in the Euler-Lagrange equations (25). Then, remarking that Λ_{ad} depends on X both explicitly and implicitly through ϵ, we have:

$$\Lambda'_{ad} = (\partial_\epsilon\Lambda_{ad})\epsilon' + \partial_X\Lambda_{ad}. \quad (54)$$

Using this relation as well as the complementarity relations on selection matrices $B^T\bar{B} = (\bar{B}B^T)^T = 0_{n_a\times n_c}$, $B^TB = 1_{n_a\times n_a}$, $\bar{B}^T\bar{B} = 1_{n_c\times n_c}$, simple projections on the spaces of allowed and constrained stress, change (25-26) into the alternative equivalent form:

$$\begin{pmatrix} g' \\ \Lambda'_c \\ \epsilon' \end{pmatrix} = \begin{pmatrix} g(\xi_o + B\epsilon)^\wedge \\ \bar{B}^T(ad_\xi^T\Lambda - \bar{F}) \\ \breve{\mathcal{H}}_r^{-1}(B^T(ad_\xi^T\Lambda - \bar{F}) - \breve{F}) \end{pmatrix}, \quad (55)$$

where $\breve{\mathcal{H}}_r = (\mathcal{H}_r + \partial_\epsilon\Lambda_{ad})$, $\breve{F} = \mathcal{H}'_r\epsilon + \partial_X\Lambda_{ad}$, ξ and Λ are given by (8) and (53) respectively, while (27) now becomes:

$$\epsilon(1) = \mathcal{H}_r(1)^{-1}(B^TF_+ - \Lambda_{ad}(1))\ ,\ \Lambda_c(1) = \bar{B}^TF_+.$$

Finally, note that (55) is now an explicit differential system with respect to (g, Λ_c, ϵ), and that such a process can be used for any continuous manipulator governed by an actuated nonlinear constitutive law of the form $B^T\Lambda = f(\epsilon, \tau(t))$, where $\partial f/\partial\epsilon$ is invertible, and $\tau(t)$ defines a set of time-varying control inputs.

B. Autonomous BVP for OCP2

1) Application of LQ optimal control theory to OCP2: A first approach, strictly consistent with Gauss principle, would consist in addressing the problem (35-38) as a LQ optimal control problem. However, it is worth noting that although $\mathcal{C}$ is quadratic with respect to the control state, it is only linear with respect to u. As a result, (46) does not provide the expression of u, and this LQ-OCP is said to be singular (see Appendix 2). Indeed, the expected dependence requires to differentiate the optimality condition of (42) twice w.r.t. to X which gives:

$$u = \mathcal{M}_{aa}^{-1}[(\Lambda_{ad} + \mathcal{H}_r\epsilon)'' - B^T(D\dot{\eta} + E\Lambda + I)], \quad (56)$$

where $\mathcal{M}_{aa}$, D, E, and I are X-dependent matrices defined in Appendix (see eq.(107)). Using (56) in (45) provides an autonomous BVP. In contrast to the static case, this BVP is linear with respect to $(\dot{\eta}, \Lambda)$ and it could be solved with some specific methods as the sweep method [26], which avoids the iterations of the shooting method by postulating the existence of a linear relationship between state and costate:

$$\forall X \in [0,1]\ :\ \Lambda(X) = S(X)\dot{\eta}(X) + s(X), \quad (57)$$

where S and s define a field of 6×6 matrix and 6×1 vector respectively, both solutions of a set of Riccati ODEs given in Appendix 2. In spite of its apparent simplicity (due to linearity), this approach suffers from several drawbacks. First, the singular nature of the OCP requires further derivations of the constitutive law, which can be numerically difficult to implement. Second, (56) is ill-conditioned which makes the Riccati equations unstable. At last, the approach provides $\ddot{\xi}_a$, which then would require to be integrated with an explicit time-integrator (see remark 5), i.e. with integration schemes that are known to be less stable (than implicit schemes), when applied to the simulation of nonlinear structural dynamics.

Remark 5: D'Eleuterio and Damaren have shown in [25] that the Newton-Euler forward dynamics algorithm of Featherstone [40], which allows to find the joint accelerations of a rigid multibody system from the knowledge of its joint state and torque variables, is the solution of a LQ problem similar to the above one, but discrete instead of continuous. This is in fact not surprising, since as mentioned earlier (remark 3), a Cosserat rod can be seen as a continuous multibody system with an infinite number of rigid bodies (the cross-sections), connected by "infinitesimal" joints (the usual vector q of joint angles being replaced by the field ξ_a). Usually, the NE forward dynamics algorithm of MBS is not deduced from optimal control, but rather by proving by induction, that a sweep-factorization of the type (57), holds from the tip body to the basis. To go a little further into details, in the discrete case, (57) becomes $\Lambda_j = S_j \dot{\eta}_j + s_j$ where the bodies index j replaces X, and where in the NE terminology [27], S_j and s_j denote the inertia matrix and the "bias-vector" of the "articulated body", whose "handle" is the body j. Based on this correspondence, one could expect to recover in $S(X)$ and $s(X)$ the continuous counterparts of these discrete concepts. Unfortunately, this is not the case because of the singular nature of OCP2. In contrast, we will see in section VI, that the resolution of the inverse dynamic problem does not suffer of the same difficulties, and that in this case, the correspondence between the NE inverse dynamic algorithm of MBS and that provided by the resolution of (45-48) when ξ_a is imposed through a prescribed time-law, is entirely preserved.

2) Regularisation of OCP2 by using an implicit time-integrator: In order to circumvent the difficulties raised by the singularity, one can directly use an implicit integration scheme from the beginning. Such an integration scheme is applied to the field ξ_a, and takes the generic form:

$$\xi_{a,n+1} = a\,\ddot{\xi}_{a,n+1} + f_n \ , \ \dot{\xi}_{a,n+1} = b\,\ddot{\xi}_{a,n+1} + h_n, \quad (58)$$

where a, b are two time-step dependent scalars, while f_n and h_n are some functions of the values of $(\xi_a, \dot{\xi}_a, \ddot{\xi}_a)$ at past steps $t_n, t_{n-1}...$, all these scalars and functions being defined by the specific scheme adopted. Now, reminding that $\ddot{\xi}_a = u$ in OCP2, the integrator (58) imposes to any $(\xi_a, \dot{\xi}_a)$ possibly solution of the dynamic balance at t_{n+1}, to fulfil the constraints:

$$\xi_a = au + f_n \quad , \quad \dot{\xi}_a = bu + h_n,$$

Moreover, thanks to the relations (8) between strains and the allowed X-rate ξ_a, one has:

$$\epsilon = au + f_n - \xi_{ao} \ , \ \dot{\epsilon} = bu + h_n \ , \ \ddot{\epsilon} = u. \quad (59)$$

Therefore, ϵ now depends on u, and the problem becomes regular since the optimality condition (46) can be explicitly used to express $u = \ddot{\epsilon}$:

$$\begin{aligned} &B^T\Lambda - \Lambda_{ad} = \mathcal{H}_r(au + f_n - \xi_{ao}) \ \Rightarrow \\ &u = \tfrac{1}{a}(\mathcal{H}_r^{-1}(B^T\Lambda - \Lambda_{ad}) - f_n + \xi_{ao}). \end{aligned} \quad (60)$$

Finally, thanks to the implicit integration and the constitutive law, the OCP becomes solvable. However, in contrast to the previous case, since $\xi = B\epsilon + \xi_o$ and $\dot{\xi} = B\dot{\epsilon}$, now depend on u through (59), the continuum kinematics (49) loose their status of reconstruction equations, to join those of the BVP (45), which is no longer linear, but takes the full nonlinear form:

$$\begin{pmatrix} g' \\ \eta' \\ \dot{\eta}' \\ \Lambda' \end{pmatrix} = \begin{pmatrix} g\hat{\xi} \\ -ad_\xi\eta + \dot{\xi} \\ -ad_\xi\dot{\eta} - ad_{\dot{\xi}}\eta + \ddot{\xi} \\ ad_\xi^T\Lambda + \mathcal{M}\dot{\eta} - ad_\eta^T\mathcal{M}\eta - \bar{F} \end{pmatrix}, \quad (61)$$

with BCs:

$$(g, \eta, \dot{\eta})(0) = (1_{4\times4}, 0, 0) \ , \ \Lambda(1) = F_+. \quad (62)$$

for a manipulator, or:

$$\Lambda(0) = -F_- \ , \ \Lambda(1) = F_+. \quad (63)$$

for a locomotor. In all cases, note that ξ, $\dot{\xi}$ and $\ddot{\xi}$ are removed from (61), by using the elimination relations:

$$\begin{aligned} \xi &= B\epsilon + \xi_o = B\mathcal{H}_r^{-1}(B^T\Lambda - \Lambda_{ad}) + \xi_o, \\ \dot{\xi} &= B\dot{\epsilon} = B(\tfrac{b}{a}(\mathcal{H}_r^{-1}(B^T\Lambda - \Lambda_{ad}) - f_n + \xi_{ao}) + h_n), \\ \ddot{\xi} &= B\ddot{\epsilon} = \tfrac{1}{a}B(\mathcal{H}_r^{-1}(B^T\Lambda - \Lambda_{ad}) - f_n + \xi_{ao}), \end{aligned} \quad (64)$$

which are directly deduced from (8) and (59,60). Note that as this is expected, when removing all the velocities and accelerations from this formulation applied to a manipulator, the autonomous dynamic BVP (61,62) is changed into the static BVP (51,52). In the case of a locomotor, there are not enough BCs to fix the solutions of (61) and in this primary formulation, the BVP is undetermined. However, we shall see in section 3 how this indetermination can be removed with an implicit time-integration scheme of the net motion dynamics. Finally, as in the static case, these nonlinear BVPs will be solved with the shooting method in the next section.

Remark 6: In the wake of works on simulation of towed cables and rods [28], a similar but different approach was proposed to address the issue of continuum robots simulation in [20]. In this alternative approach, the

continuous model of accelerations, i.e. the third ODE (top-down) of (61) being ignored (see remark 3), $\dot{\eta}$ in the costate ODE of (61), is removed with an implicit scheme of the form $\dot{\eta} = c\,\eta + f_{\eta,n}$ (with c and $f_{\eta,n}$, a scalar and a past-dependent function of $(\eta, \dot{\eta})$, fixed by the choice of the scheme). Otherwise, $\dot{\xi}$ is removed from the second ODE of (61) as above. This integration is hybrid since it involves strain (relative) and pause (absolute) accelerations, and hides the nature of the configuration space on which the model is based. In contrast, the above formulation based on Gauss principle, is consistently based on the strain configuration variables. Beyond these modeling considerations, we will see later that these two alternatives are not totally equivalent numerically.

Remark 7: In mechanics, an extensively used one-step implicit scheme is the Newmark implicit scheme. It has been originally developed for systems governed by Newtonian second order ODEs, and enriched along time of several adaptations (HHT, α-methods...), to tackle different issues related to nonlinear structural dynamics [41]. It is defined for any vector of generalized coordinates $q \in \mathbb{R}^n$, by (58), in which ξ_a is replaced by q and where we impose:

$$\begin{aligned} a = \beta\Delta t^2 \quad , \quad b = \gamma\Delta t, \\ f_n = q_n + \Delta t\dot{q}_n + \Delta t^2\left(\tfrac{1}{2} - \beta\right)\ddot{q}_n, \\ h_n = \dot{q}_n + \Delta t(1-\gamma)\ddot{q}_n, \end{aligned} \tag{65}$$

with Δt is the time step, and (β, γ) are two constant parameters tuning stability and dissipation respectively, while $(\beta, \gamma) = (1/4, 1/2)$ ensures second order accuracy with no damping, a choice that will be systematically adopted in the following.

Remark 8: Finally, the above regularization of OCP2 is based on the inversion of the constitutive law and the use of an implicit integrator of strain accelerations. Now, one can easily reflect the consequences of these two ingredients on the starting cost function. To this end, it suffices to introduce the first of the relations (59) in the term $\ddot{\epsilon}^T\mathcal{H}_r\epsilon$ of (35) or (36), to realize that this regularization consists in adding to the Gauss cost functional, a quadratic term with respect to control of the form:

$$\mathcal{C}_r = a u^T\mathcal{H}_r u, \tag{66}$$

where a is proportional to Δt^2 (for the Newmark scheme $a = \beta\Delta t^2$). Therefore, as soon as $\mathcal{H}_r = 0$, as for a continuous rigid chain, the problem becomes again singular and the approach aborts. For similar reasons, decreasing Δt also put the approach in trouble, and there exists a lower bound of Δt, noted Δt_c ("c" for "critical"), beyond which the method becomes unfeasible. From the point of view of optimal control, in such cases, the cost does not penalize enough the control which can take infinite values. This limit of the approach depends on the numerical methods used to solve the OCP, the machine accuracy, and the physical parameters of the rod as this will be illustrated later.

3) Implicit time-integration of a locomotor: In the case of a locomotor, the root cross-section $X = 0$ is free to move in space and the BVP (61,63) is undetermined, since it consists of $4 \times 6 = 24$ independent ODEs for $2 \times 6 = 12$ BCs. To remove this indetermination, it suffices to time integrate the dynamic of $g(0)$ that we note g_0, with an implicit time-integrator. As in the case of strains (see (58)), such an integrator allows the poses and velocities $(g, \eta)(0) = (g_0, \eta_0)$ to be deduced from the accelerations $\dot{\eta}(0) = \dot{\eta}_0$, and so the 3×6 kinematic ODEs of (61) require only 6 BCs. Then, since the remaining ODEs on Λ require 6 BCs, the 12 BCs (63) are sufficient in number to ensure BVP resolution. This integrator can be designed without resorting to any coordinate chart of $SO(3)$. To that end, we first replace the root cross-section pauses g_0 of $SE(3)$, by their orientation and position $(R_0, r_0) \in SO(3) \times \mathbb{R}^3$, considered as a Lie group with internal composition law $\circ$, such that $(R_1, r_1) \circ (R_2, r_2) = (R_1R_2, r_1 + r_2)$. Then, we use the usual Newmark scheme on linear spaces for the positional component r_0, i.e.:

$$r_{0,n+1} = a\,\ddot{r}_{0,n+1} + f_n \ , \ \dot{r}_{0,n+1} = b\,\ddot{r}_{0,n+1} + h_n, \tag{67}$$

with a, b, f_n, h_n defined by formulas (65) in which q is replaced by r_0, while for R_0, we use an extension of the same scheme on $SO(3)$ originally proposed in [15]:

$$\Theta_{0,n+1} = a\dot{\Omega}_{0,n+1} + k_n \ , \ \Omega_{0,n+1} = b\dot{\Omega}_{0,n+1} + l_n, \tag{68}$$

with (a, b) given by (65), and:

$$\begin{aligned} k_n = \Delta t\Omega_{0,n} + \Delta t^2(\tfrac{1}{2} - \beta)\dot{\Omega}_{0,n}, \\ l_n = \Omega_{0,n} + \Delta t(1-\gamma)\dot{\Omega}_{0,n} \ , \end{aligned} \tag{69}$$

and where the vector $\Theta_{0,n+1}$ is defined at any time step by:

$$R_{0,n+1} = R_{0,n}\exp(\hat{\Theta}_{0,n+1}), \tag{70}$$

with "exp" denoting the exponential map of $SO(3)$. Note that this scheme can be simply deduced from the usual Newmark integrator in vector space defined by (67) and (65). To that end, it suffices to make appear the displacement $r_{0,n+1} - r_{0,n}$ between two steps in the first of the relations (67), and then when shifting from $\mathbb{R}^3$ to $SO(3)$, to change this displacement into the translation along the one-parameter subgroup of $SO(3)$ defined by (70), while keeping all velocities and accelerations expressed in the mobile frame, consistently with left-invariance of rigid body dynamics on $SO(3)$ [42]. Now, removing the index $n+1$, we have the kinematic relations between $SE(3)$ and $SO(3) \times \mathbb{R}^3$:

$$\begin{aligned} g_0 = \begin{pmatrix} R_{0,n}\exp(\hat{\Theta}_0) & r_0 \\ 0_{1\times 3} & 1 \end{pmatrix}, \\ \eta_0 = \begin{pmatrix} \Omega_0 \\ R_0^T\dot{r}_0 \end{pmatrix} \ , \ \dot{\eta}_0 = \begin{pmatrix} \dot{\Omega}_0 \\ R_0^T\ddot{r}_0 + (R_0^T\dot{r}_0)\times\Omega_0 \end{pmatrix}. \end{aligned} \tag{71}$$

Finally, introducing (67) and (68) in these relations, allows expressing $(g_0, \eta_0, \dot{\eta}_0)$ at any time beyond t_n, as some

functions of $\nu_0 = (\Theta_0^T, r_0^T)^T \in \mathbb{R}^6$ only (the past values of pause, velocity and accelerations being fixed):

$$\eta_0 = A(\nu_0) \ , \ \dot{\eta}_0 = B(\nu_0) \ , \ g_0 = C(\nu_0), \qquad (72)$$

where the third of these relations is simply given by (71), while the two others are detailed in Appendix 3. Note that, as with any implicit scheme, these relations can be used to transform the differential equations of motion (here the net motions), into nonlinear algebraic equations of positional variables only, that can be treated at each time-step of a simulation, with an iterative root-finder as the Levenberg-Marquardt (LM) algorithm.

C. Shooting method for OCP1 and OCP2

To solve the autonomous BVPs (51,52) and (61-64), we apply the shooting method, which allows to solve a BVP through a sequence of initial value problems (IVPs) whose solution converges toward that of the BVP. Practically, one applies an initial guess of the unknown BCs at $X = 0$ (proximal end) that is corrected until the integrated solution of the ODEs of the BVP initialized with these proximal BCs (which so defines an IVP), matches with the known (distal) BCs at $X = 1$. This is achieved in the correction loop of the LM algorithm that formally reads, with σ the adaptive damping:

$$x^{k+1} = x^k + \left((JJ^T)(x^k) + \sigma 1\right)^{-1} J^T(x^k) r_{es}(x^k), \quad (73)$$

where k stands for the index of the iteration, x is the vector of the unknown proximal BCs, $r_{\text{es}}(x)$ the residual vector of the known distal BCs, $J = (\partial r_{\text{es}}/\partial x)$ its Jacobian, x and r_{es} being both related by the numerical forward integration (i.e. from $X = 0$ to 1), of the X-ODEs of the BVP.

Subsequently, the method is applied to the BVP of OCP1, as well as to that of OCP2 for a manipulator and for a locomotor. Analysing the known and unknown BCs of each of these cases, and using the Newmark scheme on $SO(3) \times \mathbb{R}^3$ for a locomotor, allows to instantiate x and $r_{\text{es}}(x)$ according to table 3. Going into further details, to

OCP	Statics: OCP1	Dynamics: OCP2 (manip.)
Unknown BCs	$x = \Lambda(0)$	$x = \Lambda(0)$
Residual vector	$r_{\text{es}} = \Lambda(1) - F_+$	$r_{\text{es}} = \Lambda(1) - F_+$

OCP	Dynamics: OCP2 (locom.)
Unknown BCs	$x = \nu_0$
Residual vector	$r_{\text{es}} = \Lambda(1) - F_+$

TABLE III: Table of instantiation of the shooting method for our three OCPs.

calculate the residual vector of OCP1, one first integrates (51) from $X = 0$ to 1, starting with initial conditions:

$$(g, \Lambda)(0) = (1_{4\times4}, x), \qquad (74)$$

and then compute $r_{\text{es}}(x) = \Lambda(1) - F_+$. The residual of OCP2 (manip.) is calculated by integrating (61,64) from $X = 0$ to 1, starting from:

$$(g, \eta, \dot{\eta}, \Lambda)(0) = (1_{4\times4}, 0, 0, x), \qquad (75)$$

and compute $r_{\text{es}}(x) = \Lambda(1) - F_+$. For the OCP2 (locom.), we do the same, but starting with initial conditions compatible with the constraints (72) imposed by the implicit integrator (67-70), i.e.:

$$(g, \eta, \dot{\eta}, \Lambda)(0) = (C(x), A(x), B(x), -F_-), \qquad (76)$$

and computing at the distal end: $r_{\text{es}}(x) = \Lambda(1) - F_+$. Finally, at any time-step (or "loading-step" in statics), once the LM loop has converged (i.e. $\|r_{\text{es}}\|$ is below a given threshold), ξ in statics, and $(\xi, \dot{\xi}, \ddot{\xi})$ in dynamics, are updated with (64). In dynamics, they are stored for the next step, in f_n and h_n of (65). The "loading" or time-step is then incremented, and the shooting algorithm resumes. The initial guess of this algorithm can be defined as follows. For a manipulator, $x = \Lambda(0)$ is initialized by its value at the previous loading or time-step, while for a locomotor, one can use one of the usual predictors of the Newmark scheme as the inertial (ballistic) one, which consists in choosing $(r_{0,n+1}, \dot{r}_{0,n+1})$ and $(\Theta_{0,n+1}, \Omega_{0,n+1})$ such that $\ddot{r}_{0,n+1}$ and $\dot{\Omega}_{0,n+1}$ are forced to zero in (67) and (68) respectively.

To complete the picture, the LM loop (73) needs the Jacobian of the residual vector $J = (\partial r_{\text{es}}/\partial x)$, to be calculated. This can be achieved by first linearizing each of the IVP, to get their tangent IVP, that we note here TIVP. We now give these TIVP for OCP1, OCP2 (manip.) and OCP2 (locom.). The first one is given by the linearization (or second variation Δ) of (25-27), the second and the third by that of (61-64). Note that all these TIVPs are fed with different initial perturbative conditions at $X = 0$ that do not affect X in any case (i.e. $\Delta X = 0$). Therefore, the second variation Δ behaves as the first one δ, and fulfills the commutation relation (5). Defining $\Delta\zeta$ such that $\Delta g = g\Delta\hat{\zeta}$, the TIVPs are defined as follows:

- The TIVP of OCP1 is defined by gathering (51) and:

$$\begin{pmatrix} \Delta\zeta' \\ \Delta\Lambda' \end{pmatrix} = \begin{pmatrix} -ad_\xi\Delta\zeta + \Delta\xi \\ ad_\xi^T\Delta\Lambda + ad_{\Delta\xi}^T\Lambda - \Delta\bar{F}_{\text{ext}} \end{pmatrix}, \qquad (77)$$

where from (64), and since $\Delta\Lambda_{ad} = 0$:

$$\Delta\xi = B\mathcal{H}_r^{-1}B^T\Delta\Lambda, \qquad (78)$$

and with ICs given by (74), and:

$$(\Delta\zeta, \Delta\Lambda)(0) = (0, \Delta x). \qquad (79)$$

- The TIVP of OCP2 is defined by gathering (61) and:

$$\begin{pmatrix} \Delta\zeta' \\ \Delta\eta' \\ \Delta\dot{\eta}' \\ \Delta\Lambda' \end{pmatrix} = \begin{pmatrix} -ad_\xi\Delta\zeta + \Delta\xi \\ -ad_\xi\Delta\eta - ad_{\Delta\xi}\eta + \Delta\dot{\xi} \\ -ad_\xi\Delta\dot{\eta} - ad_{\dot{\xi}}\Delta\eta \\ \mathcal{M}\Delta\dot{\eta} - ad_\eta^T\mathcal{M}\Delta\eta \end{pmatrix}$$

$$+\begin{pmatrix} 0 \\ 0 \\ -ad_{\Delta\dot{\xi}}\eta - ad_{\Delta\xi}\dot{\eta} + \Delta\ddot{\xi} \\ -ad^T_{\Delta\eta}\mathcal{M}\eta + ad^T_{\xi}\Delta\Lambda + ad^T_{\Delta\xi}\Lambda - \Delta\bar{F}_{\text{ext}} \end{pmatrix}, \quad (80)$$

with $\Delta\xi$, $\Delta\dot{\xi}$, and $\Delta\ddot{\xi}$, given by the variation of (64):

$$\Delta\xi = B\mathcal{H}_r^{-1}B^T\Delta\Lambda \ \ , \ \ \Delta\dot{\xi} = \frac{b}{a}\Delta\xi \ \ , \ \ \Delta\ddot{\xi} = \frac{1}{a}\Delta\xi. \quad (81)$$

And with ICs given by (75), and:

$$(\Delta\zeta, \Delta\eta, \Delta\dot{\eta}, \Delta\Lambda)(0) = (0, 0, 0, \Delta x). \quad (82)$$

in the case of a manipulator, and by (76) and:

$$(\Delta\zeta, \Delta\eta, \Delta\dot{\eta}, \Delta\Lambda)(0) = \quad (83)$$
$$((\partial C/\partial x)\Delta x, (\partial A/\partial x)\Delta x, (\partial B/\partial x)\Delta x, -\Delta F_-).$$

in the case of a locomotor, where we used the tangent maps (Jacobian matrices) to (72) detailed in Appendix 3. Finally, integrating these linearized systems of ODEs from $X = 0$ to 1, and keeping only the variation of the known BCs in $X = 1$, provides $\Delta r_{\text{es}} = \Delta\Lambda(1) - \Delta F_+$, for each of them, where ΔF_+, like ΔF_- and $\Delta\bar{F}$, can be calculated from the variation (4) of a possibly state dependent model of the tip wrench F_+ (resp. F_- and $\bar{F}$). By simple identification of this numerical computation with the explicit matrix relation $\Delta r_{es} = (\partial r_{\text{es}}/\partial x)\Delta x$, one can infer the following computational process of $J = (\partial r_{\text{es}}/\partial x)$. Feeding the above linear ODEs with unit initial conditions $\Delta x = \delta_i$, $i = 1, 2, ..., 6$, with δ_i a vector of zero entries, except the i^{th} which is fixed to 1, allows the i^{th} column of the Jacobian to be computed. Thus resuming this process column after column, completely fills the matrix.

VI. Inverse statics and dynamics of continuum robots

In the above development, we addressed the forward static and dynamic models, which can be formally defined as the input-output maps: $\xi_a = FSM(\text{state}, \Lambda_{ad})$ and $\ddot{\xi}_a = FDM(\text{state}, \Lambda_{ad})$ respectively, where in statics the state of the rod is defined by g or ξ_a (through integration of $g' = g\hat{\xi}$), and in dynamics by $(\xi_a, \dot{\xi}_a)$, or equivalently (g, η) thanks to the reconstruction equations (49). Inverting the role of inputs and outputs defines the inverse static and dynamic models: $\Lambda_a = ISM(\text{state}, \xi_{ad})$ and $\Lambda_a = IDM(\text{state}, \ddot{\xi}_{ad})$ respectively. Solving these inverse problems is much more simple than solving their forward counterparts. Indeed, in this case, $u = \epsilon$ is known in (25), and $u = \ddot{\epsilon}$ is known in (45), which so directly define two autonomous BVPs.

A. Inverse statics of a continuum manipulator

In the static case, it suffices to impose $\xi_a = \xi_{ad}$ in the state ODE of (25) that is integrated forward, and to reintroduce g (which appears in $\bar{F}(g)$ and $F_+(g)$) in the co-state equation which is integrated backward from $X = 1$, where $\Lambda(1) = F_+(g)$, to $X = 0$. This provides the field Λ along the entire rod, which is then used to calculate $\Lambda_{ad} = B^T\Lambda - \mathcal{H}_r\epsilon$. Note that this algorithm in two passes is the continuum version of the Luh computed torque algorithm (here in statics) of rigid manipulators [43].

B. Inverse dynamics of continuum robots

In the dynamic case, we need to find the two fields $(\dot{\eta}, \Lambda)$ solution of the system of ODEs (45) with BCs (47) or (48) where $u = \ddot{\xi}_a = \ddot{\xi}_{ad}$ while other state-dependent fields are considered as functions of X only (e.g. updated at each time t of a time loop fed by a prescribed motion $t \mapsto (\xi_{ad}, \dot{\xi}_{ad}, \ddot{\xi}_{ad})(t)$). This problem defines an autonomous linear BVP with state and costate $(\dot{\eta}, \Lambda)$ in the generic form (108) of Appendix 2. As introduced in section V.B, one can then apply the backward sweep method, which in both cases (manipulator or locomotor), consists in exploiting the factorization (57). Then, using the identification and elimination process introduced in Appendix 2, one obtains that S and s are solutions of the Riccati ODEs:

$$S' = \mathcal{M} + ad^T_\xi S + S ad_\xi,$$
$$s' = ad^T_\xi s - \tilde{F} + S(ad_{\dot{\xi}}\eta - B\ddot{\xi}_{ad}), \quad (84)$$

where we introduced the notation $\tilde{F} = \bar{F} + ad^T_\eta\mathcal{M}\eta$. These two ODEs need to be X-integrated backward with initial conditions $S(1) = 0$ and $s(1) = \Lambda(1) = F_+$. Once S and s known in $X = 0$, if the robot is a locomotor, $\dot{\eta}(0) = \dot{\eta}_0$ is an unknown of the problem that one has to compute from (57) in $X = 0$ and the first of the BCs of (48):

$$\Lambda(0) = -F_- = S(0)\dot{\eta}_0 + s(0)$$
$$\Rightarrow \dot{\eta}_0 = -S(0)^{-1}(s(0) + F_-). \quad (85)$$

If the robot is a manipulator, the same process can be applied but directly with $\dot{\eta}(0) = 0$. Finally, this resolution can be implemented through the following (inverse) algorithm which proceeds in three passes. First integrate forward (from 0 to 1) the reconstruction equations (49), then integrate backward (from 1 to 0) the two Riccati's ODEs (84), starting with initial conditions $S(1) = 0$ and $s(1) = \Lambda(1) = F_+$. Once S and s known over $[0, 1]$, one can calculate $\dot{\eta}$ by forward integrating the state equation of (45) initialised with $\dot{\eta}(0) = 0$, for a manipulator, and (85), for a locomotor. While integrating the state, calculate the costate field Λ with (57) an the actuated stress $\Lambda_{ad} = B^T\Lambda - \mathcal{H}_r\epsilon$, which is the output of the inverse dynamics.

Remark 9: This inverse algorithm is new, and in particular different from, though equivalent to, the one in [8]. When $\mathcal{H}_r = 0_{n_a \times n_a}$, it is nothing more than a continuous version of the inverse algorithm of MBS based on the composite bodies [27]. As its discrete homologous, it is structured into similar functional steps, where the 3 (2 forward and 1 backward) recursions on body indices of the discrete case, are replaced by 3 ODEs (2 forward and 1 backward) on the continuous label X. In the terminology of MBS [27], $S(X)$ and $s(X)$, solutions of (84), are the exact

continuous counterpart of the inertia matrices and external-inertia wrenches of the composite bodies of a MBS, i.e., they define the inertia matrix and wrench with respect to the X-cross-sectional frame, of a composite beam constituted of all the cross-sections included into $[X, 1]$ and frozen in their current configuration. Going further, using standard relations between ad and Ad operators [44], allows integrating (84), which yields:

$$S(X) = \int_1^X Ad_h^{-T} \mathcal{M} Ad_h^{-1} dY \ , \ s(X) = Ad_{h(1)}^{-T} F_+ - \int_1^X Ad_h^{-T} (\tilde{F} + S(B\ddot{\xi}_{ad} - ad_{\dot{\xi}}\eta)) dY, \quad (86)$$

where $h = g^{-1}(X)g(Y)$ here represents the transformation mapping the X-cross-sectional frame onto the Y-one ($Y \in [X, 1]$). Examining (86) confirms that $S(X)$ and $s(X)$ represent the sum of all the cross-sectional inertia matrices and external-inertia wrenches along the rigid piece of beam $[X, 1]$, consistently carried from their local Y-frame to the X-frame.

Remark 10: As formulated above, the inverse problem is related to the input-output map $\Lambda_{ad} \rightarrow \xi_a$ in statics, and $\Lambda_{ad} \rightarrow \ddot{\xi}_a$ in dynamics. In practise, these inverse static and dynamic models need to be supplemented with the inversion of the map $\tau \rightarrow \Lambda_{ad}$, where τ is a finite set of actuation inputs that depends on the technology of the actuation. For example, in the case of TACRs, $\tau = (T_1, T_2, ...T_N)^T$ represents the vector of tendon tensions in the equation (17), and this additional map can be inverted by generalized inversion as illustrated in [17].

VII. Numerical applications

To illustrate the above simulation approach, we consider four tests. We start with a tendon-actuated continuum manipulator statically bent by pulling one tendon (test 1), and suddenly released in gravity (test 2). In addition to providing a validation against other simulators, this second test allows to discuss the behavior of the dynamic simulator. The third test is inspired of the bench of the flying beam initially proposed by Simo to validate the model of rigid overall motions in his geometrically exact FEM (GE-FEM) [15]. This third test allows to validate our geometric integrator on $SO(3) \times \mathbb{R}^3$ as well as the OC-based shooting method applied to a rod free at both ends. In a last example (test 4), the approach is applied to the simulation of an undulatory swimmer at high Reynolds numbers internally actuated with a torque field. All the results are compared to those provided by another (totally different) simulator recently validated against GE-FEM in [17]. In this simulator, the strains are projected on a truncated basis of Legendre polynomials (modes), whose coefficients are governed by Lagrangian dynamics coupled to those of the pause of the root (parameterized with quaternions), all being explicitly time-integrated with ODE45 of Matlab. The same integrator (ODE45) is here used to X-integrate the IVPs and TIVPs of the shooting method. As regards time-integration, we use the previously introduced Newmark schemes with $(\beta, \gamma) = (1/4, 1/2)$ which ensures second order accuracy and no damping.

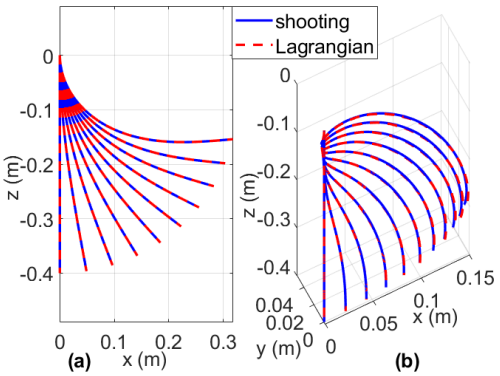


Fig. 2: A rod is deformed by pulling one tendon of a convergent (a) and spiral (b) pair. Snapshots of static equilibrium configurations obtained by solving OCP1 with the shooting method applied to the BVP (51,52).

A. Test 1: a TACR statically deformed

We consider a single piece TACR of length $l = 0.4$m and disk diameters $2R_d = 0.04$m, equipped with a pair of convergent and spiraled tendons [17]. Initially aligned with gravity (i.e. with the vertical), it is deformed by pulling one of the two tendons (indexed by $1, 2$), in the quasi-static regime. Simulations are carried out by applying the shooting method to the autonomous BVP (51,52) of OCP1 with the actuation model (17) in which $\Gamma = E_1 = (1, 0, 0)^T$, the influence of ϵ is neglected, while $D_1 = (0, R_d(1-(X/l)), 0)^T$ and $D_1 = (0, R_d \cos(2\pi X/l), R_d \sin(2\pi X/l))^T$ (1^{st} convergent and spiral routing). Figure 2 displays the converged equilibrium configurations when incrementing the pulling force T_1 step by step from 0 to 67.5N. The arm is modelled by a rod with $\rho = 8000$kg.m^{-3} and $E = 207$GPa, which are typical of continuum robots [20]. In these examples, which are representative of all others here not reported, the two simulators fit exactly in the first case, and very well (error at the tip $\simeq 1.4\%$), in the second. Numerically, the shooting method just requires a few iterations of the LM algorithm of Matlab `fsolve`.

B. Test 2: TACR bent and released in gravity

We reconsider the TACR of test 1 in static equilibrium under the effect of a convergent tendon with a tension T_1. At $t = 0$s, the tendon is instantly cut and the rod is released under the effect of its stiffness and gravity. Solving the OCP2 of this manipulator at each time step, allows to simulate its temporal evolution. Figure 3(a) shows the time plots of the tip of the rod, here released from a weakly deformed static configuration ($T_1 = 15$N). The simulation is performed over 1s, with $\Delta t = 1$ms. Once again, the results obtained with the shooting method (in blue), here applied to the BVP (61,62), are very close to

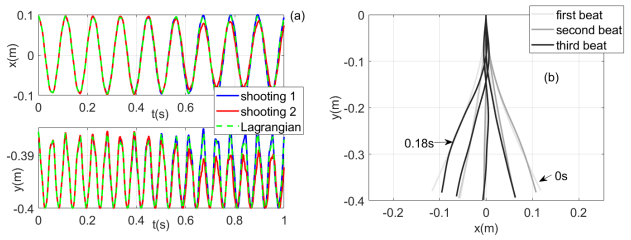


Fig. 3: Rod of test 1 pulled and released from a weakly deformed configuration: (a) Time-evolution of the tip position obtained with the shooting method applied to the BVP of OCP2 with (blue), and without acceleration equation (red), and with the reference simulator (green). (b) Snapshots illustrating the numerical instability due to $\Delta t < \Delta t_c^\star$.

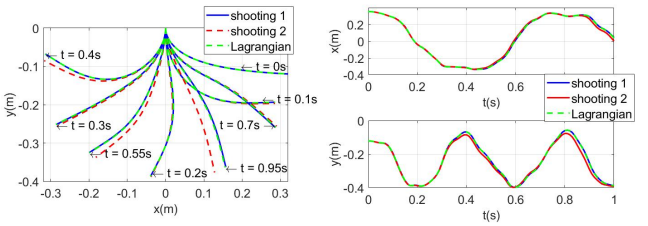


Fig. 4: Rod of test 1 with $k = k_\star/100$, pulled with an horizontal force and released from a highly deformed configuration: Snapshots (a), and time-evolution of the tip coordinates (b). The color code is the same as in figure 3.a.

those provided by the Lagrangian simulator of [17] (green) which here needs 5 modes to achieve convergence. Beyond validation, the same example was tested under many other conditions in order to obtain an overview of the approach, which can be summarized as follows. First, we remarked that the shooting method is much more difficult to apply in dynamics than in statics and requires good initial guesses to converge. Otherwise, the IVPs defined by (61) can blow up before reaching the distal end. These difficulties increase in the regime of finite deformations and/or strong transients, and the method cannot generally be applied in a single shoot, but rather in several ones using the "multiple" or "modified shooting method" of [45]. In all that follows, we have used a non adaptive version of the latter approach with 10 intermediate shoots. Second, as mentioned in remark 8, decreasing the time-step Δt tends to degrade the numerical convergence of the LM algorithm. Calculating the Jacobian of the residual vector with the TIVPs as explained in section V.C, contributes to delay divergence, but there always exists a critical time-step Δt_c below which the method fails. In order to estimate the value of Δt_c, we started by considering the rod of test 1. For the sake of concision, we use the notations $(EI, \rho A, l) = (k, \mu, l)$, while any quantity related to this reference rod is indicated with a star. Based on successive numerical trials and errors, we noticed that the numerical instability of the simulator is not a threshold instability, as can be a standard CFL condition [46], but rather an instability with a "transitional" regime. After a finer analysis, we obtained that $0.5\text{ms} \lesssim \Delta t_c^\star \lesssim 1\text{ms}$. From this first result, we tried to determine how Δt_c varies when the physical parameters (k, μ, l) change. The results of this study led us to the following scaling relationships:

$$\frac{\Delta t_c}{\Delta t_c^\star} \sim \sqrt{\frac{k_\star}{k}}, \quad \frac{\Delta t_c}{\Delta t_c^\star} \sim \sqrt{\frac{\mu}{\mu_\star}}, \quad \frac{\Delta t_c}{\Delta t_c^\star} \sim \left(\frac{l}{l_\star}\right)^2, \tag{87}$$

which suggest that Δt_c follows a law of the form:

$$\Delta t_c \propto \chi\, l^2 \sqrt{\frac{\rho A}{EI}}, \tag{88}$$

where χ is a dimensionless pre-factor. It seems quite natural that Δt_c evolves in this way because we recognize in the factor of χ, the characteristic time of a linear Euler-Bernoulli beam [47]. In this regard, note that such a result could also have been obtained by a dimensional analysis involving only the parameters k, μ and l. It can be seen, however, that the pre-factor χ seems to depend on other physical factors as the conditions of the test (imposed forces and motions), and also non-physical ones as the adopted numerical solver, and the machine accuracy. For example, in the conditions of the test considered here, the χ factor can be estimated at $1/100$. It is also worth noting that the method proposed here is based on an OCP which tends to be singular when the matrix $a\mathcal{H}_r$ of (66) becomes ill-conditioned (see remark 8). In the present context $a\mathcal{H}_r$ is of the order of $EI\Delta t^2$, which suggests once again that Δt_c evolves inversely with the square root of $k = EI$, as the first relation (87) does. We also noted that Δt_c is little influenced by the amplitude of the deformations. Figure 3(b) shows a few snapshots (every 15ms), of a simulation of the star rod released from a weakly deformed configuration with $\Delta t = 0.5\text{ms} < \Delta t_c^\star$. Around 0.18s, the rod seems loosing its stiffness and some fast undulations appear at its distal tip and increase up to destabilize simulation. Based on remark 6, we also applied the same shooting approach but to the BVP defined by removing from (61) the continuous model of accelerations (the third ODE from top to bottom). To achieve comparison in equivalent conditions of time-integration, we used a one-step Newmark-like implicit approximation of time-derivatives as proposed in [28], i.e., we approximate $\dot{\epsilon}$ and $\dot{\eta}$ in (61) by:

$$\dot{\epsilon} = c\,\epsilon + f_{\epsilon,n} \quad , \quad \dot{\eta} = c\,\eta + f_{\eta,n}, \tag{89}$$

where $c = (\alpha\Delta t)^{-1}$, $f_{z,n} = \alpha(\alpha-1)^{-1}\dot{z}_n - c\,z_n$, for $z = \epsilon$ and $z = \eta$, and with α a parameter that tunes numerical damping. In the results here reported we took $\alpha = 1/2$, which ensures no damping [28], as this is the case of the Newmark-integrator (58,65). Note that in this case, (89) also coincides with the trapezoidal approximation of [20], which has a second order accuracy like our Newmark integrator. In Figure 4, we have plotted the results of a 1s simulation (snapshots and time plots of the tip position), performed with both simulators based on shooting, and with the Lagrangian approach of [17]. The rod is the reference (star) rod with a material one hundred times softer

($k = k^\star/100$). It is initially bent with a horizontal tip force of 0.4N, and released at $t = 0$s. As observed in these plots (which are representative of all others performed in the regime of finite deformations), the full BVP-based shooting approach fits perfectly with the Lagrangian simulator, while the second one (based on the BVP without the accelerations model) shows small discrepancies. Note that such differences also appear in the small deformation regime, but after many beats (see red plots of figure 3(a)). Despite this, the deformations of both approaches remain in good agreement with the reference simulator. Another key point of comparison between the two approaches is that they are constrained by critical time steps of almost identical values, suggesting that the interpretation of Δt_c based on Gauss least constraint principle and OCT also holds for the second shooting approach.

C. Test 3: a flexible flying stick

This test is directly inspired by the GE-FEM bench of the flying rod, recently simulated by applying the Lagrangian approach of [17]. In this test, a straight rod is initially tilted as illustrated in figure 5.a. Gravity is ignored, and at $t = 0$s, a force and a torque are applied to one of its tips ($X = 1$) according to the time-profiles of figure 5.a, which are entirely quantified by $c_{2,\max}$. The stick is thus catapulted in space in a complex three-dimensional movement. The rod here considered has the material properties of the slender swimmer tested in the next example (1m length, 0.1m diameter, $\rho = 10^3$kg.m^{-3}, $E = G = 10^6$Pa). In Figures 5.b and c, we plotted the time evolution of the coordinates of the center of mass of the stick and a sequence of snapshots every 0.2s for a 5s simulation with $c_{2,\max} = 10.7$Nm. It is obtained with the shooting method applied to the BVP (61,63) with the Kirchhoff rod model. As shown in these graphs, the results obtained with the shooting-based approach (here with $\Delta t = 10\text{ms} > \Delta t_c \simeq 3\text{ms}$) are in very good agreement with those of the simulator of [17], which thus confirms its applicability to three-dimensional floating basis systems, time-integrated with our implicit Newmark-scheme on $SO(3) \times \mathbb{R}^3$.

D. Test 4: a continuum bio-inspired swimmer

We now apply the approach to a bio-inspired locomotor, namely a continuum swimmer mimicking a fish. The fish body is modelled as a neutrally buoyant 1m length Kirchhoff rod composed of elliptical cross sections as detailed in [48] and illustrated in figure 6. The hydrodynamic forces are modelled with Lighthill's theory [49], in the geometric setting of [48],[50]. In this context, we use the change of variable on stress:

$$\Lambda^\star = V_1 \mathcal{M}_a \eta - \begin{pmatrix} 0_{3\times 1} \\ T_f E_1 \end{pmatrix} \ , \ T_f = \frac{1}{2}\eta^T \mathcal{M}_a \eta, \qquad (90)$$

where V_1 is the first (axial) component of $\dot{r}$ in the cross-sectional frame, $E_1 = (1,0,0)^T$, while $\mathcal{M}_a$ stands for the 6×6 added mass matrix of the elliptic cross-sections, and

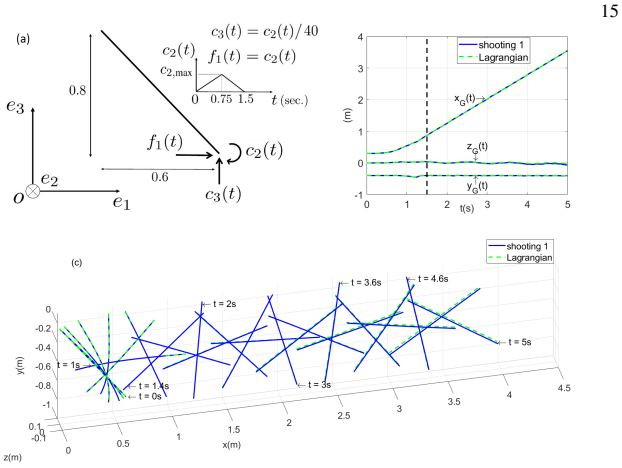


Fig. 5: Flying stick: (a) Conditions of the test. (b) Mass center coordinates vs time. (c) Snapshots over 5s. Blue/green: shooting/Lagrangian based simulator.

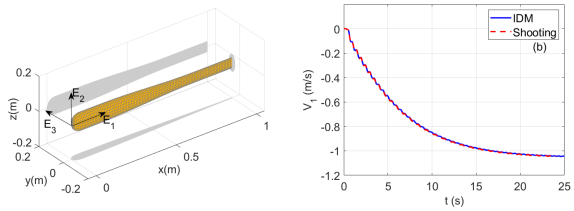


Fig. 6: (a) Morphology of the swimmer. (b) $V_1(t)$ given by the FDM (red/shooting) and the IDM (blue/Riccati ODEs).

T_f is the corresponding density of kinetic energy along the body. With these notations, the dynamic model of the swimming fish is given by that of a Cosserat rod (in vacuum) in which Λ needs to be replaced by $\Lambda^\star$, $\mathcal{M}$ by $\mathcal{M}^\star = \mathcal{M} + \mathcal{M}_a$, and with the BCs:

$$\Lambda^\star(0) = 0 \ , \ \Lambda^\star(1) = \left[V_1 \mathcal{M}_a \eta - \begin{pmatrix} 0_{3\times 1} \\ T_f E_1 \end{pmatrix} \right] (1). \quad (91)$$

To generate a swimming gait, we take an active constitutive law (16) in the form of the distributed proportional control:

$$\Lambda_a = \Lambda_{a,d} + \mathcal{H}_r \xi_a = \mathcal{H}_r (\xi_a - \xi_{a,d}), \qquad (92)$$

where $(X,t) \in [0,1] \times \mathbb{R}^+ \mapsto \xi_{a,d}(X,t) \in \mathbb{R}^{n_a}$ defines a desired time-evolution of the shape. In the test here reported, we consider a planar swimming gait defined by $\xi_{a,d}(X,t) = (0, K_{d,2}, 0)(X,t)$ with:

$$K_{d,2}(X,t) = f_r(t) f_w(X,t), \qquad (93)$$

where we used two functions. One is a sinusoid ramp time-function f_r of the form:

$$f_r(t) = 0 \ , 0 \le t < t_i \ , \ f_r(t) = 1 \ , t \ge t_f, \qquad (94)$$
$$f_r(t) = \frac{t - t_i}{t_f - t_i} - \frac{1}{2\pi} \sin\left(2\pi \frac{t - t_i}{t_f - t_i} \right) \ , t_i \le t < tf,$$

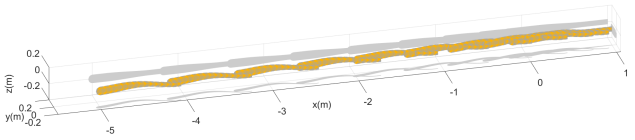


Fig. 7: Snapshots of a continuum undulatory swimmer torque-controlled with an active constitutive law.

which is twice continuously differentiable and thereby guarantees smooth time transitions. The second, is a space-time (backward) wave function f_w defined $\forall (X,t) \in [0,1]\times\mathbb{R}^+$:

$$f_w(X,t) = (a_0+a_1X+a_2X^2)\sin\left[2\pi\left(\frac{X}{\lambda}-\frac{t}{T}\right)\right], \quad (95)$$

in which λ is the wave length, T is the period, and a_0, a_1 and a_2 are the coefficients of a second order polynomial amplitude envelop. To simulate this swimmer, we applied the shooting algorithm to the BVP (61,63) in which Λ is replaced by $\Lambda^\star$ and $F_\pm$ defined by (91). We assume the fish has no stiffness except that of the control law (92). Noting from (90) and $B = (1_{3\times3}, 0_{3\times3})^T$ (the fish is modelled by a Kirchhoff rod), that $B^T\Lambda = B^T\Lambda^\star$, the elimination relations are still (64), in which $\Lambda_{ad} = -\mathcal{H}_r\xi_{a,d}(t)$, $\xi_{a,o} = 0_{3\times1}$. To illustrate the approach on a representative example, we reported in figures 6(b) and 7, the time evolution of the head axial velocity $V_1(X=0)$ (first component of $\dot{r}(0)$ in the mobile frame of the cross-section $X=0$), and a sequence of snapshots obtained with the shape law (93-95) with $(a_0,a_1,a_2) = (1,0.5,2)$, $(t_i,t_f) = (0,1)$, $(\lambda,T) = (1,1)$ (SI Units). The time-step of the test is $\Delta t = 10\text{ms} > \Delta t_c \simeq 1\text{ms}$. The control stiffness along the body is $\mathcal{H}_r = \text{diag}(0,50,0)$, which is enough to ensure ξ_a to perfectly track ξ_{ad}. Hence, as illustrated in figure 6(b), the motion of figure 7 matches with that obtained by directly shape-controlling the swimmer with the inverse algorithm of section VI.B, i.e. by time-integrating (85) with ODE45, where $S(0)$ and $s(0)$ are computed at each time step, by X-integrating the Riccati ODEs (84).

VIII. Conclusion

In this paper, we have studied the relationship between modeling and simulation of continuous robots and optimal control theory. Based on the Cosserat model, we have shown that beyond the static case of passive rods actuated by external forces, one can integrate in the optimal control theory the modeling and simulation of continuous robots actuated in a distributed manner, both in the quasi-static and dynamic regimes. In statics, this extension is based on the minimum potential energy principle applied to a Cosserat rod in which the actuation is modeled by an internal stress field through an extended active constitutive law. In dynamics, it is achieved by invoking Gauss's principle of least constraint, a variational principle rarely used in rational mechanics, more oriented towards Hamilton's and D'Alembert's principle. Using these two principles, the statics and dynamics of continuous manipulators and bioinspired locomotors have been formulated into OCP based BVPs that have been solved at each step of a simulation loop with the shooting method, as this is generally done in the continuum robotics community. Remarkably, the OC perspective has shown that while the static optimal control problem is well-posed, the dynamic problem is inherently singular. In the context of numerical simulation, this singularity can be circumvented by using an implicit time integrator and the invertibility of the constitutive law. However, this regularization process imposes a certain limit to the time-step, below which the approach fails. Although this phenomenon has been noticed as one of the major difficulties of the shooting-based approach applied to dynamics, it has so far been interpreted as a numerical artifact of the method due to the rounding errors of the machine. On the contrary, the OC viewpoint shows that this limitation of the method is in fact due to the intrinsic structure of the dynamic OCP. In addition, OC also shows that the dynamic BVP must contain the continuous acceleration model, which is systematically omitted in the shooting-based approaches used so far to solve the forward dynamics of Cosserat rods. As our numerical tests show, adding this model to the dynamic BVP improves the accuracy of the simulations. Moreover, as a by-product of this improvement, the full BVP provided by OC turns to be a continuous version of the Newton-Euler model of rigid robots, thus linking the practices of continuous robotics with the discrete dynamics of rigid robots. Beyond the case of manipulators, the entire approach was extended to locomotors and applied to a continuum swimmer inspired by slender fish. In the context of bio-inspired robotics and biomechanics, this simulator is a first step towards studying the interactions of musculoskeletal and nervous systems with the ambient flow, on which is based the emergent synchronization of undulatory swimming [51]. Finally, let's end this article with some practical tips for the users. In statics, the shooting-based approach is efficient and always gives satisfaction. However, in dynamics, it is much more fragile and fails when the time-step is smaller than a critical value defined by (88). As shown in this relationship, this limitation becomes more and more restrictive as the robot becomes softer and softer, and for this reason we could not recommend using it (at least in its current state of the art) in soft robot dynamics. Despite this limitation, the shooting-based approach has several advantages. First, it does not require any reduction of the configuration space and, from this point of view, should be very accurate. Second, it is conceptually simple to program, at least for simple architectures (like those simulated in the paper). Third, the dimension of its residual vector is very small and independent of the space discretization, and when it works, the method is very time-efficient.

IX. Appendix 1: Calculus of the Gauss Constraint for a Continuum Robot

To calculate the acceleration energy of a continuum robot, we calculate the contribution dT_{acc} of a single rigid cross-

section noted $\mathcal{S}(X)$, and add all these contributions along the rod axis. Any material point of the rod is localized by its material coordinates (X, Y, Z) where X is the label of the cross-section containing the particle, and (Y, Z) are its coordinates in the plane of the cross-section. Then using the definition (30), we have:

$$dT_{\text{acc}} = \frac{1}{2}\int_{\mathcal{S}(X)} \rho\, a_c^2\, dY dZ, \tag{96}$$

where $dYdZ$ is the area element of the section $\mathcal{S}(X)$, and $a_c : (X,Y,Z) \in \mathbb{R}^3 \mapsto a_c(X,Y,Z) \in \mathbb{R}^3$ is the acceleration field of the material points of the rod. For the sake of concision, we introduce the vector $X_\perp = (0, Y, Z)^T$ such that $(X,Y,Z)^T = (X,0,0)^T + X_\perp$. According to Gauss principle, a_c needs to be compatible with the Cosserat kinematics, i.e., it can be detailed as:

$$a_c(X,Y,Z) = \ddot{\varphi}(X,Y,Z) = \tag{97}$$
$$R(X)(\dot{V}(X) + \hat{\dot{\Omega}}(X)X_\perp + \hat{\Omega}(X)(V(X) + \hat{\Omega}(X)X_\perp)),$$

where $\ddot{\varphi}$ is deduced by time-differentiating twice the Cosserat kinematics $\varphi(X,Y,Z) = r(X) + R(X)X_\perp$. In these conditions, one has:

$$dT_{\text{acc}} = \tfrac{1}{2}\textstyle\int_{\mathcal{S}(X)} \rho\, \ddot{\varphi}^2 dA = \tag{98}$$
$$\tfrac{1}{2}\textstyle\int_{\mathcal{S}(X)} \rho((\dot{V} + \hat{\dot{\Omega}}X_\perp) + \hat{\Omega}(V + \hat{\Omega}X_\perp))^2 dY dZ,$$

which can be developed as:

$$\begin{aligned} dT_{\text{acc}} &= \tfrac{1}{2}\textstyle\int_{\mathcal{S}(X)} \rho(\dot{V} + \hat{\dot{\Omega}}X_\perp)^2 dY dZ \\ &+\tfrac{1}{2}\textstyle\int_{\mathcal{S}(X)} \rho(\hat{\Omega}(V + \hat{\Omega}X_\perp))^2 dY dZ \\ &+ \textstyle\int_{\mathcal{S}(X)} \rho(\dot{V} + \hat{\dot{\Omega}}X_\perp)^T(\hat{\Omega}(V + \hat{\Omega}X_\perp)) dY dZ \\ &= C_1 + C_2 + C_3. \end{aligned} \tag{99}$$

We now compute each of the integrals of (99), noted C_1, C_2 and C_3. Regarding C_1, let us remark that the ordinary kinetic energy of a cross-section being defined by:

$$dT = \tfrac{1}{2}\textstyle\int_{\mathcal{S}(X)} \rho\, \dot{\varphi}^2 dY dZ =$$
$$\tfrac{1}{2}\textstyle\int_{\mathcal{S}(X)} \rho(V + \hat{\Omega}X_\perp)^2 dY dZ = \tfrac{1}{2}\eta^T \mathcal{M}\eta, \tag{100}$$

we have for the same reasons (i.e. using the same calculations but with accelerations instead of velocities):

$$C_1 = \frac{1}{2}\dot{\eta}^T \mathcal{M}\dot{\eta}. \tag{101}$$

Regarding C_2, one can ignore it, since it only depends on velocities and brings no contribution to dT_{acc}. It remains to calculate the double product C_3. It can be detailed first as:

$$C_3 = \int_{\mathcal{S}(X)} \rho(\dot{V} + \hat{\dot{\Omega}}X_\perp)^T(\hat{\Omega}(V + \hat{\Omega}X_\perp)) dY dZ. \tag{102}$$

Once more, developing this expression while ignoring the terms that only depend on velocities, and assuming that the cross-sectional frame is centered on the mass-center of the cross-section (i.e. $\int_{\mathcal{S}(X)} \rho\, X_\perp dY dZ = 0$), one finds:

$$C_3 = \textstyle\int_{\mathcal{S}(X)} \rho\, \dot{V}^T(\hat{\Omega}V) dY dZ +$$
$$\textstyle\int_{\mathcal{S}(X)} \rho(\hat{\dot{\Omega}}X_\perp)^T(\hat{\Omega}(\hat{\Omega}X_\perp)) dY dZ. \tag{103}$$

where using the definition of the mass $\rho A = \int_{\mathcal{S}(X)} \rho\, dY dZ$ and inertia matrix of the cross section $\rho J = \int_{\mathcal{S}(X)} \rho \hat{X}_\perp \hat{X}_\perp^T dY dZ$, one can show that:

$$\begin{aligned} C_3 &= \dot{V}^T(\Omega \times \rho A V) + \dot{\Omega}^T(\Omega \times \rho J \Omega) \\ &= -\dot{\eta}^T(ad_\eta^T \mathcal{M}\eta), \end{aligned} \tag{104}$$

where we used the definition of ad_η and $\mathcal{M} = \text{diag}(\rho J, \rho A 1_{3\times 3})$. Finally, accounting for (101) and (104), with $C_2 = 0$, in (99), and integrating dT_{acc} along the robot, provides the expression (30) of its "energy of acceleration". To calculate the acceleration power of non reactive force P_{acc}, we first remark that these forces are: the external forces applied on the two tips (F_-, F_+) and inside the domain of the rod $(\bar{F})$, those generated by the exogenous activation (Λ_{ad}), and finally the internal restoring forces. Summing the usual definitions of power of all these forces, and changing velocities into accelerations, gives (31,32), where remind that the usual (velocity) power of internal restoring forces is $P_{\text{int}} = -\frac{d}{dt}\left(\frac{1}{2}\int_0^1 \epsilon^T \mathcal{H}_r \epsilon dX\right) = -\int_0^1 \dot{\epsilon}^T(\mathcal{H}_r\epsilon)dX$.

X. Appendix 2: Singular LQ-OCP2

Even if the LQ optimal problem of the forward dynamics is singular and indeed numerically unfeasible due to the ill-conditioning of the Riccati equations (see section V.B.1), we here ignore these numerical considerations and formally address this problem. When applied to a cost functional quadratic w.r.t. state and control, the optimality condition (42-c) directly provides the expected expression of the control and the optimal control is said to be regular. In contrast, in our case, the optimal cost $\mathcal{C}$ is only linear w.r.t. the control, and our problem is singular. To derive the expected expression of $\ddot{\xi}_a = u$, we need to differentiate (42-c) further w.r.t. to X, up to make appear explicitly the control [26]. As proved in [52], the number of derivation required is even and noted $2p$, where p defines the order of the singular arcs along which the state and costate evolve. Applying this process to our problem, gives for the zero and first order X-derivatives of (42-c):

$$\begin{aligned} \left(\tfrac{\partial H}{\partial u}\right) = 0 \Rightarrow B^T\Lambda &= \Lambda_{ad} + \mathcal{H}_r\epsilon\ , \\ \left(\tfrac{\partial H}{\partial u}\right)' = 0 \Rightarrow B^T\Lambda' &= \Lambda'_{ad} + (\mathcal{H}_r\epsilon)', \end{aligned} \tag{105}$$

which does not make appear the control, since from (45.b): $\Lambda' = \mathcal{M}\dot{\eta} + ad_\xi^T\Lambda - \tilde{F}$. In contrast, because Λ'' makes appear $\dot{\eta}'$, and finally $\dddot{\xi}_a = u$ through the state ODE (45.a), the second order X-derivative does provide the expected dependence:

$$\begin{aligned} \left(\tfrac{\partial H}{\partial u}\right)'' = 0 \Rightarrow B^T\Lambda'' &= \Lambda''_{ad} + (\mathcal{H}_r\epsilon)'' = \\ &\mathcal{M}_{aa}u + B^T(D\dot{\eta} + E\Lambda + I), \end{aligned} \tag{106}$$

where we introduced the notations:

$$\begin{aligned} &\mathcal{M}_{aa} = B^T\mathcal{M}B \quad , \quad D = \mathcal{M}' + ad_\xi^T\mathcal{M} - \mathcal{M}ad_\xi\ , \\ &E = ad_{\xi'}^T + ad_\xi^T ad_\xi^T\ , \ I = -\tilde{F}' - ad_\xi^T\tilde{F} - \mathcal{M}ad_{\dot{\xi}}\eta, \\ &\tilde{F} = \bar{F} + ad_\eta^T\mathcal{M}\eta. \end{aligned} \tag{107}$$

As a result, the singular arcs are of degree one, and the expression of u provided by inverting (106), is defined by (56). Then, using (56) in (45) provides the autonomous BVP (indexes s and c mean "state" and "costate" respectively):

$$\begin{pmatrix} \dot{\eta}' \\ \Lambda' \end{pmatrix} = \begin{pmatrix} P_{ss} & P_{sc} \\ P_{cs} & P_{cc} \end{pmatrix} \begin{pmatrix} \dot{\eta} \\ \Lambda \end{pmatrix} + \begin{pmatrix} p_s \\ p_c \end{pmatrix}, \quad (108)$$

where we introduced the notations:

$$P = \begin{pmatrix} P_{ss} & P_{sc} \\ P_{cs} & P_{cc} \end{pmatrix} = \begin{pmatrix} -ad_\xi - GD & -GE \\ \mathcal{M} & ad_\xi^T \end{pmatrix},$$

$$p = \begin{pmatrix} p_s \\ p_c \end{pmatrix} = \begin{pmatrix} B\mathcal{M}_{aa}^{-1}(\Lambda_{ad} + \mathcal{H}_r\epsilon)'' - ad_{\dot{\xi}}\eta - GI \\ -\tilde{F} \end{pmatrix},$$

with $G = B\mathcal{M}_{aa}^{-1}B^T$. But all the entries of P and p, only depend on $(\xi, \dot{\xi})$ and not of the state $\dot{\eta}$. As a result, they can be considered as X-dependent functions, and this BVP is linear with respect to $(\dot{\eta}, \Lambda)$. As such, it can be solved with the "sweep method" [26], which postulates the linear relationship between state and costate (57). Now, identifying $\Lambda' = (S\dot{\eta} + s)'$ with the costate equation of (108), and introducing into this identity, the state equation of (108), as well as the sweep relation (57), one finds:

$$\begin{aligned} &S'\dot{\eta} + S(P_{ss}\dot{\eta} + P_{sc}(S\dot{\eta} + s) + p_s) + s' \\ &\quad = P_{cs}\dot{\eta} + P_{cc}(S\dot{\eta} + s) + p_c. \end{aligned} \quad (109)$$

But such a relation needs to be true for any $\dot{\eta} \neq 0$, which provides the Riccati ODEs for $S(.)$ and $s(.)$:

$$\begin{aligned} S' &= P_{cs} - SP_{ss} + P_{cc}S - SP_{sc}S, \\ s' &= P_{cc}s + p_c - SP_{sc}s - Sp_s. \end{aligned} \quad (110)$$

Finally, the LQ resolution of OCP2 would progress as follows. First integrate forward (from 0 to 1) the reconstruction equations (49), then integrate backward (from 1 to 0) the two Riccati's ODEs, starting with initial conditions $S(1) = 0$ and $s(1) = \Lambda(1) = F_+$. Once S and s known over $[0, 1]$, one can calculate $\dot{\eta}$ by forward integrating the state equation of (108), while calculating the costate field Λ with (57). For a manipulator, this last integration is merely initialised with $\dot{\eta}(0) = 0$, while for a locomotor, we use $\dot{\eta}(0) = S^{-1}(0)(\Lambda(0) - s(0)) = -S^{-1}(0)(F_- + s(0))$, which requires the invertibility of $S(0)$.

XI. Appendix 3: Expressions of $C(\nu_0)$, $A(\nu_0)$, $B(\nu_0)$ and their Jacobians

While $C(\nu_0)$ is directly given by (71), $A(\nu_0)$, $B(\nu_0)$ can be detailed as (remind that $\nu_0 = (\Theta_0^T, r_0^T)^T$):

$$A(\nu_0) = \begin{pmatrix} \Omega_0 \\ V_0 \end{pmatrix} = \begin{pmatrix} \tilde{a}\Theta_0 + \tilde{k}_n \\ R_0^T(\tilde{a}r_0 + \tilde{f}_n) \end{pmatrix}, \quad (111)$$

$$B(\nu_0) = \begin{pmatrix} \dot{\Omega}_0 \\ \dot{V}_0 \end{pmatrix} = \begin{pmatrix} \tilde{b}\Theta_0 + \tilde{l}_n \\ R_0^T(\tilde{b}r_0 + \tilde{h}_n) + V_0 \times \Omega_0 \end{pmatrix}$$

where $R_0^T = \exp(\hat{\Theta}_0)^T R_{0,n}^T$, and $\tilde{b} = a^{-1}$, $\tilde{a} = b\tilde{b}$, $\tilde{h}_n = -\tilde{b}f_n$, $\tilde{f}_n = h_n - \tilde{a}f_n$, $\tilde{l}_n = -\tilde{b}k_n$, $\tilde{k}_n = l_n - \tilde{a}k_n$. Differentiating the above expressions as well as (71), one can show that:

$$\frac{\partial C}{\partial \nu_0} = \begin{pmatrix} T(\Theta_0) & 0_{3\times3} \\ 0_{3\times3} & R_0^T \end{pmatrix}, \quad (112)$$

$$\frac{\partial A}{\partial \nu_0} = \begin{pmatrix} \tilde{a}1_{3\times3} & 0_{3\times3} \\ \hat{V}_0 T(\Theta_0) & \tilde{a}R_0^T \end{pmatrix}, \quad (113)$$

$$\frac{\partial B}{\partial \nu_0} = \begin{pmatrix} \tilde{b}1_{3\times3} & 0_{3\times3} \\ (\hat{A}_0 - \hat{\Omega}_0\hat{V}_0)T(\Theta_0) + \tilde{a}\hat{V}_0 & (\tilde{b}1_{3\times3} - \tilde{a}\hat{\Omega}_0)R_0^T \end{pmatrix},$$

where $A_0 = \dot{V}_0 + \Omega_0 \times V_0$, and $(R_0^T \Delta R_0)^\vee = T(\Theta_0)\Delta\Theta_0$, is the differential of the exponential of $SO(3)$ [15].

XII. Acknowledgment

This publication is supported by the French ANR COSSE-ROOTS under Grants ANR-20-CE33-0001, and partly by the Khalifa University of Science and Technology under Grants CIRA-2020-074, RC1-2018-KUCARS. The first author would like to thank his colleague Georges Levey for their fruitful discussions on the direct dynamics of Cosserat beams in the early 2000s.

References

[1] V. Jurdjevic, "Non-euclidean elastica," *American Journal of Mathematics*, vol. 117, no. 1, pp. 93–124, 1995.

[2] ——, *Geometric Control Theory*, cambridge ed. Cambridge University Press, 1997.

[3] T. Bretl and Z. McCarthy, "Quasi-static manipulation of a kirchhoff elastic rod based on a geometric analysis of equilibrium configurations," *The International Journal of Robotics Research*, pp. –, 2013.

[4] G. Walsh, R. Montgomery, and S. Sastry, "Optimal path planning on matrix lie groups," in *Proceedings of 1994 33rd IEEE Conference on Decision and Control*, vol. 2, 1994, pp. 1258–1263 vol.2.

[5] V. Rao Anil, "A survey of numerical methods for optimal control," *Advances in the Astronautical Sciences*, vol. 135, no. 1, pp. 497–528, 2009.

[6] P. D. O'Reilly, O.M., "On stability analyses of three classical buckling problems for the elastic strut." *Journal of Elasticity*, vol. 105, pp. 117–136, 2011.

[7] G. A. Bliss, *Lectures on the calculus of variations*. Chicago Univ. Press, 1947.

[8] 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.

[9] 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.

[10] P. E. Dupont, J. Lock, B. Itkowitz, and E. Butler, "Design and control of concentric-tube robots," *IEEE Trans. Robot.*, vol. 26, no. 2, pp. 209–225, apr 2010.

[11] D. C. Rucker, B. A. Jones, and R. J. Webster, "A geometrically exact model for externally loaded concentric-tube continuum robots," *IEEE Trans. Robot.*, vol. 26, no. 5, pp. 769–780, oct 2010.

[12] 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.

[13] 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.

[14] A. D. Marchese and D. Rus, "Design, kinematics, and control of a soft spatial fluidic elastomer manipulator," *The International Journal of Robotics Research*, 2015.

[15] 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.

[16] M. Bergou, M. Wardetzky, S. Robinson, B. Audoly, and E. Grinspun, "Discrete elastic rods," in *ACM transactions on graphics (SIGGRAPH)*. ACM, 2008.

[17] F. Boyer, V. Lebastard, F. Candelier, and F. Renda, "Dynamics of continuum and soft robots: A strain parameterization based approach," *IEEE Transactions on Robotics*, 2020.

[18] J. Ha, F. Park, and P. Dupont, "Elastic stability of concentric tube robots subject to extemal loads," *IEEE Transactions on Biomedical Engineering*, vol. 63, no. 6, pp. 1116–1128, 2016.

[19] J. Till and D. Rucker, "Elastic stability of Cosserat rods and parallel continuum robots," *IEEE Transactions on Robotics*, vol. 33, no. 3, pp. 718–733, 2017.

[20] 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.

[21] 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," *Proceedings of the National Academy of Sciences of the United States of America*, vol. 107, pp. 19 832–19 837, 2010.

[22] C. F. Gauss, "Uber ein neues allgemeines grundgesetz der mechanik," in *J. Reine Agnew. Math.*, vol. 4, 1829, pp. 232–235.

[23] J. W. Gibbs, "On the fundamental formulae of dynamics," *American Journal of Mathematics*, vol. 2, no. 1, pp. 49–64, 1879.

[24] P. Appell, "Sur une forme générale des équations de la dynamique," in *C. R. Acad. Sci., Paris*, vol. 129, 1899, pp. 459–460.

[25] G. M. T. D'Eleuterio and C. J. Damaren, "The relationship between recursive multibody dynamics and discrete-time optimal control," *IEEE Transactions on Robotics and Automation*, vol. 7, no. 6, pp. 743–749, 1991.

[26] A. E. Bryson and Y. C. Ho, *Applied Optimal Control*. London: Taylor and Francis, 1975.

[27] R. Featherstone, *Rigid Body Dynamics Algorithms*. Springer New York, 2008.

[28] Y. Sun, "Modelling and simulation of low-tension oceanic cable/body deployment," *PHD dissertation, University of Connecticut*, 1996.

[29] H. Poincaré, "Sur une forme nouvelle des equations de la mecanique," *Compte Rendu de l'Academie des Sciences de Paris*, vol. 132, pp. 369 – 371, 1901.

[30] J. E. Marsden and T. S. Ratiu, *Introduction to Mechanics and Symmetry, 2nd ed.* Springer New York, 1999.

[31] F. Boyer and F. Renda, "Poincare's equations for cosserat media: Application to shells," *Journal of Nonlinear Science*, 2016.

[32] 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.

[33] J. E. Marsden and T. S. Ratiu, *Introduction to Mechanics and Symmetry*, 2nd ed. Springer-Verlag, 1999.

[34] H. Wakamatsu, E. Arai, and S. Hirai, "Knotting/unknotting manipulation of deformable linear objects," *The International Journal of Robotics Research*, vol. 25, no. 4, pp. 371–, 2006.

[35] M. Levy, *Classical mechanics with optimal control and variational calculus, an intuitive introduction*. American Mathematical Society, 2014.

[36] P. Appell, *Traité de mécanique rationnelle*. Gauthier-Villars et Cie., Paris, 1931.

[37] M. M. Athans and P. L. Falb, *Optimal Control*. New York: Dover Publication, 1966.

[38] F. Boyer, S. Ali, and M. Porez, "Macrocontinuous dynamics for hyperredundant robots: Application to kinematic locomotion bioinspired by elongated body animals," *IEEE Transactions on Robotics*, vol. 28, pp. 303–317, 2012.

[39] 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.

[40] 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.

[41] A. Cardona and M. Geradin, "Time-integration of the equations of motion in mechanism analysis," *Computers and Structures*, vol. 33, no. 3, pp. 801–820, 1989.

[42] V. I. Arnold, *Mathematical methods in classical mechanics*, 2nd ed. Springer-Verlag, New-York, 1988.

[43] M. W. Walker, J. Y. S. Luh, and R. C. P. Paul, "On−line computational scheme for mechanical manipulator," *Transaction ASME, J. of Dyn. Syst., Measurement and Control*, vol. 102, no. 2, pp. 69–76, 1980.

[44] R. Murray, Z. Li, and S. Sastry, *A Mathematical Introduction to Robotic Manipulation*. Taylor & Francis, Boca Raton, USA, 1994.

[45] R. Holsapple, R. Venkataran, and D. Doman, "A modified simple shooting method for solving two-point boundary-value problems," in *IEEE Aerospace Conference Proceedings*, May 2003, pp. 2783–2790.

[46] R. Courant, K. Friedrichs, and H. Lewy, "On the partial difference equations of mathematical physics," *Mathematische Annalen*, vol. 100, pp. 32–74, 1928.

[47] L. Meirovitch, *Dynamics and Control of structures*. Wiley, New York, 1989.

[48] F. Boyer, M. Porez, and A. Leroyer, "Poincare 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.

[49] M. J. Lighthill, "Aquatic animal propulsion of high hydromechanical efficiency," *Journal of Fluid Mechanics*, vol. 44, pp. 265–301, 11 1970.

[50] F. Boyer, M. Porez, A. Leroyer, and M. Visonneau, "Fast dynamics of an eel-like robot-comparisons with navier-stokes simulations," *IEEE Trans. Robot.*, vol. 24, no. 6, pp. 1274–1288, Dec. 2008.

[51] R. Thandiackal, K. Melo, L. Paez, J. Herault, T. Kano, K. Akiyama, F. Boyer, D. Ryczko, A. Ishiguro, and A. Ijspeert, "Emergence of robust self-organized undulatory swimming based on local hydrodynamic force sensing," *Science Robotics*, vol. 6, no. 57, 2021.

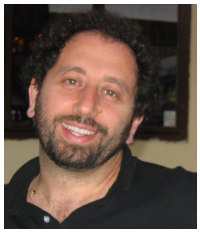
[52] B. S. Goh, "Necessary conditions for singular extremals involving multiple control variables," *J. SIAM Control*, vol. 4, pp. 716–731, December 1966.

Frédéric Boyer was born in France in 1967. He received the Diploma in mechanical engineering from the Institut Nationale Polytechnique de Grenoble, Grenoble, France, in 1991, the Master of Research degree in mechanics from the University of Grenoble in 1991, and the Ph.D. degree in robotics from the University of Paris VI, Paris, France, in 1994. He is currently a Professor with the Department of Automatic Control, IMT-Atlantique, Nantes, France, where he is a member of the Robotics Team, Laboratoire des Sciences du Numérique de Nantes (LS2N). His current research interests include structural dynamics, geometric mechanics, and biorobotics (locomotion dynamics and underwater electric sensing). Dr. Boyer received the Monpetit Prize from the Academy of Science of Paris in 2007 for his work in dynamics and the French "La Recherche Prize" in 2014, for his works on artificial electric sense. He has coordinated several national projects and one European FP7-FET project on a reconfigurable eel-like robot able to navigate with electric sense.

Vincent Lebastard joined the École Normale Supérieure de Cachan, France, in 2001, and received an aggregation from the ministry of the Education in Electrical Engineering in 2002. He received a Ph.D degree from the University of Nantes, France, in 2007. He is currently a Assistant Professor with the Department of Automatic Control, IMT-Atlantique, Nantes, France, where he is a member of the Robotics Team, Laboratoire des Sciences du Numérique de Nantes (LS2N). Dr. lebastard received the French "La Recherche Prize" in 2014, for his works on artificial electric sense. His research interests include biorobotics and dynamic modeling.

Dr Fabien candelier received both the Diploma degree in mechanical engineering and the Master of Research degree in Fluid Mechanics from the University of Lorraine (UHP Nancy, France). He completed his PhD in Fluid Mechanics in 2005 at the Institut National Polytechnique de Nancy, Nancy. He was an assistant professor at the Ecole des Mines de Nantes during three years prior to joining the University of Aix-Marseille, as a lecturer (Maitre de Conférence). He is a fluid dynamicist, with particular interest in modeling. His research mainly focuses on the dynamics of spherical or anisotropic particles and on the locomotion of slender bodies (Lighthill theories).

Dr. Federico Renda received his B.Sc. and M.Sc. degrees in Biomedical Engineering in 2007 and 2009, respectively, from the University of Pisa. He completed his Ph.D. in Robotics in 2014 from Scuola Superiore Sant'Anna. Since 2015, he has been a Post-Doctoral Fellow with the Khalifa University Robotics Institute (KURI) at Khalifa University, where he presently serves as Assistant Professor in the Department of Mechanical Engineering. His research interests include dynamic modeling and control of soft and continuum robots using principles of geometric mechanics. Dr. Renda is also a member of the Institute of Electrical and Electronics Engineers (IEEE).



Mazen Alamir is research director at CNRS, France. He graduated in Mechanics (Grenoble, 1990) and Avionics (Toulouse 1992). He received his Ph.D. in Nonlinear Model Predictive Control in 1995 from Grenoble Institute of Technology. He served in the Control Systems Department of Gipsa-lab, Grenoble where he served as head of the Nonlinear Systems and Complexity research group. His main research topics are model predictive control, moving-horizon observers, nonlinear hybrid systems and signature-based diagnosis and control of biological systems. He was a member of the IFAC technical committee on Nonlinear Systems as well as the IEEE Conference Editorial Board and still serves as Associate Editor of the IEEE Transaction on Automatic Control. He is co-founder and scientific advisor of the startup Amiral-Technologies, specialized in AI algorithms for industrial predictive maintenance and diagnosis. He is author of about 250 Publications including about 90 peer reviewed journal papers, two monographies and thirteen industrial patents.