A Geometrically-Exact Assumed Strain Modes Approach for the Geometrico- and Kinemato-static Modellings of Continuum Parallel Robots

Sébastien Briot, Frédéric Boyer

A Geometrically-Exact Assumed Strain Modes Approach for the Geometrico- and Kinemato-static Modellings of Continuum Parallel Robots

Sébastien Briot[1] and Frédéric Boyer[2]

***Abstract*—There is a growing interest on the study of continuum parallel robots (CPRs) due to their higher stiffness and better dynamics capacities than serial continuum robots (SCRs). Several works have focused on the computation of their geometrico- and kinemato-static models, that can be sorted into two main categories: (i) models based on the continuous Cosserat equations: They are very accurate but assessing elastic stability with them is tricky; (ii) discretized models: They allow easily checking the elastic stability but they require a large number of elastic variables to be accurate.**

In this paper, we extend an approach based on assumed strain modes developed for the dynamics of SCRs to the statics of CPRs. This method is able to predict the robot configuration with an excellent accuracy with a very limited number of elastic variables, contrary to other discretization methods. The method is also more than 100 times faster than finite differences for a better prediction accuracy. Finally, it is possible to assess the robot elastic stability by only checking the Hessian of the potential energy as for any discretization method, thus making the analysis of this property simpler than for the continuous Cosserat model. All results are validated through simulations on two case studies.

***Index Terms*—Continuum parallel robots, Modelling, Statics, Assumed modes, Stability**

NOMENCLATURE OF THE MAIN SYMBOLS

In the present paper, in order to differentiate scalars from matrices (vector instantiations being considered as single-column matrices), scalars will be in italic, and matrices in bold. Bold lowercase (or eventually calligraphic letters) will be used for vectors and group transformation, bold uppercase for matrices with more than one column.

$\boldsymbol{\xi}$, $\boldsymbol{\gamma}$, $\boldsymbol{\kappa}$	Space-twist field and its linear and rotational parts, respectively.
$\boldsymbol{\eta}$, $\mathbf{v}$, $\boldsymbol{\omega}$	(Time-)twist field and its linear and rotational parts, respectively.
$\boldsymbol{\Phi}$	Strain function matrix.
$\boldsymbol{\Psi}$	Vector of constraints.
$\boldsymbol{\lambda}$	Lagrange multipliers.
$\boldsymbol{\tau}_{\mathrm{a}}$	Motor input efforts.
$\mathbf{1}_{p\times p}$	Identity matrix of dimension p.
$\mathbf{A}$, $\mathbf{P}$, $\mathbf{E}$, $\mathbf{W}$	Robot kinematic Jacobian matrices related to the variables $\mathbf{q}_{\mathrm{a}}$, $\mathbf{q}_{\mathrm{p}}$, $\mathbf{q}_{\mathrm{e}}$ and $\mathbf{w}$.
$\mathbf{Ad}$, $\mathbf{ad}$	Two operators from Lie group theory.
$\boldsymbol{g} = (\mathbf{R}, \mathbf{p})$	A group transformation of $SE(3)$, represented by an homogeneous transformation matrix composed with a rotation matrix $\mathbf{R}$ and a translational vector $\mathbf{p}$.
$\mathbf{h}$	Unit quaternion vector.
l	Length of a rod.
$\mathcal{Q}$	Vector of strain generalized forces in the static equations.
$\mathbf{q}_{\mathrm{a}}$, $\mathbf{q}_{\mathrm{p}}$, $\mathbf{q}_{\mathrm{u}}$, $\mathbf{q}_{\mathrm{e}}$	Motor, controlled, uncontrolled and elastic variables.
$\mathbf{q}$, $\mathbf{q}_{\mathrm{au}}$, $\mathbf{q}_{\mathrm{pu}}$	Vectors $[\mathbf{q}_{\mathrm{a}}^T, \mathbf{q}_{\mathrm{p}}^T, \mathbf{q}_{\mathrm{u}}^T]^T$, $[\mathbf{q}_{\mathrm{a}}^T, \mathbf{q}_{\mathrm{u}}^T]^T$ and $[\mathbf{q}_{\mathrm{p}}^T, \mathbf{q}_{\mathrm{u}}^T]^T$.
$\mathbf{w}$, $\mathbf{f}$, $\mathbf{m}$	A wrench, a force, a moment.
$\bar{\mathbf{w}}$	A distributed wrench.
$\mathbf{Z}$	A matrix spanning the null space of $\mathbf{J}$, the matrix of the kinematic constraints.

*This work was partially supported by the French ANR project COSSEROOTS (ANR-20-CE33-0001).

The authors would like to thank Federico Zaccaria for his help on the shooting method.

The authors would like to thank the unknown reviewers for their valuable remarks that greatly contributed to the improvement of the paper.

[1] S. Briot is with the Laboratoire des Sciences du Numérique (LS2N) at the Centre National de la Recherche Scientifique (CNRS), 44321, Nantes, France. Sebastien.Briot@ls2n.fr

[2] F. Boyer is with the Laboratoire des Sciences du Numérique (LS2N) at the Institut Mines Telecom Atlantique (IMTa), 44300, Nantes, France. frederic.boyer@imt-atlantique.fr

I. INTRODUCTION

Continuum robots [1] have been introduced in order to enhance the limited interaction capacities of rigid-link robot manipulators. Most of them are made with a serial architecture, composed by a serial assembly of slender rods deformed by wires [2], [3], electromagnets [4], [5], fluidic actuators [6]–[8], shape memory alloy based actuators [9], [10], electro-active polymers [11], [12] or other types of actuation (e.g. concentric tube robots [13]–[15] or also multi-backbone robots [16]–[18]).

While being of interest for many applications requiring safe human-robot interaction, like minimally invasive surgery [19], serial continuum robots have also their own limitations. Typically, they have low stiffness and limited dynamics capacities. In order to overcome these issues, the concept of continuum parallel robots (CPR) have been recently explored. CPRs were first proposed in [20], [21]. Similarly as for rigid-link parallel robots [22], CPRs are composed of several kinematic chains connected in parallel and attached to a moving platform. However, each kinematic chain is made of elastic links subject to large deformations.

There is a growing interest in CPRs and several architectures have been recently proposed, among which we may distinguish (the list is not exhaustive):

- CPRs with legs of varying lengths, like for instance in [20], [21], [23], [24] where continuum Gough-Stewart-like platforms were studied.
- CPRs with legs of constant lengths, mounted on a motor moving on the ground. Several planar CPRs of this type have been proposed in [25]–[31] and some spatial robots have been analyzed in [32]–[34].
- CPRs actuated by tendons [35]–[37].
- Reconfigurable CPRs [37], [38] in which the position or orientation of the motors can be modified.
- CPRs with legs constrained by intermediate links [39]–[42].

The geometrico-static model of these robots[1] was the main focus of several works. In [20], [21], the model was established by using the Cosserat's rod theory: This theory allows obtaining a system of nonlinear Differential Algebraic Equations (DAEs) characterizing the robot's static equilibrium. In [26]–[29] a quasi-analytical description of the robot equilibrium configurations based on the Kirchhoff's model with planar assumptions, is proposed. However, this model is valid under planar motion conditions only (spatial robots cannot be modelled) and it allows to apply wrenches on the platform only (gravity field on the legs cannot be taken into account for instance). In [35], [36], the Piecewise Constant Curvature model [43] was applied while lumped modelling and finite differences were used in [34] and in [44], respectively. More details on the modelling of Cosserat beams are provided in Section I-A.

These models can serve then in order to characterize some geometry or kinematic properties of the robot, like for instance: The robot workspace [20], [29], [31], [32], [35], [45]–[47]; The end-effector positioning error [31], [32]; Kinetostatic properties (compliance, force transmission, or manipulability) [21]; Singularities [33]. One of the most crucial property to be assessed is the robot elastic stability [33], [48]. Using technics of optimal control theory, it is shown in [48], that a robot static configuration is not necessarily a minimizer of the potential energy, and that its stability must be verified by analyzing second-order conditions based on the analysis of the non-discretized DAEs characterizing the robot deformation given by the calculus of variations. However, although mathematically exact, the method for checking the stability in [48] is based on tools from optimal control theory that are unfamiliar to the mechanical engineering culture. In further details, this analysis is based on a rather intricate stability test based on the detection of the so-called "conjugate points": If conjugate points can be detected, the robot is unstable; Otherwise it is stable. In [33], the DAEs were discretized and the stability was studied by checking the positive-definiteness of the Hessian matrix of the potential energy. However, in order to get a good pose estimation with any discretized model, the number of elements must be large (typically $>$ 1000 with finite differences), leading to big size matrices and little computational efficiency.

[1]A geometrico-static (kinemato-static, resp.) model is a model which can be obtained by using not only the geometry (kinematics, resp.) equations of the robot, but also its statics equations.

A. Methods for modelling Cosserat rods

At the crossroads of rigid body mechanics and continuum mechanics, the Cosserat rod model is an ideal tool for the study of slender bodies undergoing large deformations. Initially presented as an abstract object [49], it has been applied over time to many problems in engineering sciences such as structural mechanics, where it gave birth to the geometrically exact finite element method (GE-FEM) [50], in ocean engineering for the simulation of submarine cables [51], or in computer graphics, for the needs of interactive simulation [52]. In robotics, whether for the study of hyper-redundant bioinspired locomotion [53], the simulation and control of non-invasive continuous medical robots [54], [19], or for the design of new concepts of soft arms [55], it is gradually becoming a standard, comparable to the multi-body models of rigid robotics. In these various contexts, the exploitation of the Cosserat model can be divided into two main categories depending on whether one considers the model as fully continuous [53], [56], or whether one seeks to reduce it on a functional basis of finite dimension [55], [57]. In the first case, direct and inverse dynamic models have been proposed and applied to several issues related to simulation and control, while in the second case, beyond the PCC (Piecewise Constant Curvature) and PCS (Piecewise Constant Strain) methods [43], [55], the reduction of the strain fields on an arbitrary functional basis [57], has opened new promising perspectives for robotics.

B. Originality of the work and organization of the paper

The contributions of the article are of two kinds depending on whether one considers the Cosserat modeling viewpoint of [57] or the stability analysis of CPRs from [21] and [33].In the first case, the paper extends the modeling by strain modes to the statics of CPRs, i.e., to closed kinematic loop systems consisting of rigid bodies and Cosserat rods connected by active or passive localized joints. It should be noted that such an extension of the strain-based parameterization has already been proposed in [58] to deal with the dynamics of closed-chain soft robots. However, in [58], this extension was performed with the Piecewise Constant Strain (PCS) approach, i.e., with the first zero order components of the higher dimensional polynomial strain basis used in this paper. As a consequence of this difference, unlike in [58], the accuracy of the model proposed here is not increased by adding pieces to the bodies, in the manner of what is done for spatial discretization in FEM, but rather by increasing the dimension of a strain basis defined on a single piece along each body. Moreover, while in [58], the kinematic constraints of the closing loops are treated at the level of accelerations and used to remove their associated Lagrange multipliers in the dynamic equilibrium, our approach is purely static and exclusively based on a geometric model of kinematic loops and the static equilibria of the subsystems. As a consequence of this choice, it is more directly adapted to the usual needs of kinetostatic analysis in parallel robotics where dynamics is ignored, at least at first. Without anticipating too much on the results of the article, we will show that this is possible to accurately predict the robot configuration with a very reduced number of strain modes in comparison to other

standard discretization methods. Typically, 10 elastic variables per leg lead to a prediction accuracy of 50 microns for 1-meter-length rods. Moreover, for a same accuracy, the computational time required by the approach is similar to that based on the full continuous Cosserat model of [21], while it is 100 times faster than the simulation approach based on finite-differences used in [33]. In the second context, we show that the proposed formalism is compatible with the stability analysis framework shown in [33], but making it more efficient. Indeed, once the static model of a CPR is reduced on a basis of strain modes, it takes the usual form of Lagrangian mechanics and its elastic stability can be studied by simply analyzing the spectrum of the reduced Hessian of the potential energy [33] as this is done for any finite-dimensional model. However, in our case, the size of the Hessian matrices to be computed is much smaller than with the finite-difference approach used in the simulations of [33], making the analysis of the spectrum much more computational efficient. Beyond its conceptual simplicity, it is worth noting that the calculation of the reduced Hessian of the CPR only requires the computation of matrices already derived for the calculation of the robot kinemato-static model[2]. Thus, our formalism offers an efficient alternative approach to the optimal-control-based analysis of the full continuous model of [21], which is used in our simulations as a reference. Based on all these remarks, the main motivation of the article is to show how the static stability analysis of CPRs can be performed with the usual methods of finite dimensional mechanics.

The paper is organized as follows. The next section deals with the computation of geometrico-static model of a single rod. We reintroduce the discretization framework proposed in [57] that was used in order to obtain the dynamics model of continuum robots with serial architecture. However, we modified the approach so that we can have a computation of the Jacobian matrices relating the motion of the leg tip to the model input variables. In Section III, we first compute the geometry and kinematic constraint equations necessary in order to close the kinematic chains, and we then show how to obtain the CPR geometrico-static model based on the equations for a single rod. Then, in Section IV, the kinemato-static model is developed, and the way to verify the robot stability is detailed. Section VI presents some case studies, and finally, in Section VII, conclusions are drawn.

II. MODEL OF A COSSERAT BEAM

In this section, we apply the geometrically-exact assumed strain modes parametrization of [57] to a single rod, and propose a different numerical implementation of its kineto-static equations. This model will be next used to derive the multi-rod model of a CPR.

A. Geometry and kinematics of the beam

In what follow, for the sake of concision, we use a bit of Lie group notations [60]. We detail them in the Appendix A. Let us consider an hyperelastic beam for which we assume that displacements are finite and strains are small. In what follows, we consider that this beam will be one of the legs of a *CPR*. Therefore, we will attach a subscript i to all of its variables.

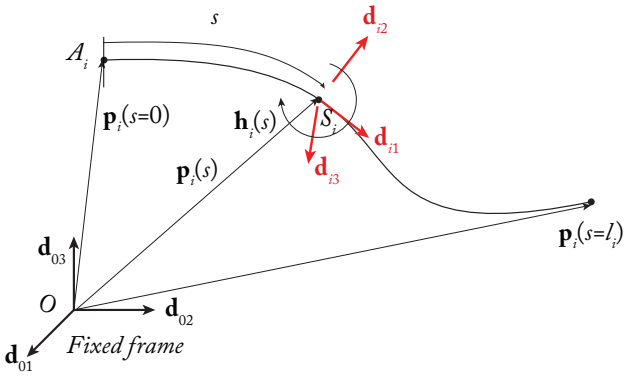


Fig. 1. Parameterization of the continuum slender rod

The Cosserat model consider that this deformable body can be modeled by a set of continuously rigid cross-sections stacked along a material line, parameterized by a curvilinear abscissa $s \in [0, l_i]$. The location of the cross-section at the abscissa s is parameterized by a frame $\mathcal{F}_i(s) = (S_i, \mathbf{d}_{i1}, \mathbf{d}_{i2}, \mathbf{d}_{i3})(s)$, where $S_i(s)$ and $\mathbf{d}_{i1}(s)$ coincide with the center of the cross section and its unit normal vector, respectively. As a result, the pose of the s-cross-section is parameterized by the action of an element $\boldsymbol{g}_i \in SE(3)$ on the reference frame $\mathcal{F}_0 = (O, \mathbf{d}_{01}, \mathbf{d}_{02}, \mathbf{d}_{03})$: $\boldsymbol{g}_i(s) = (\mathbf{R}_i, \mathbf{p}_i)(s)$, where $\mathbf{R}_i(s) \in SO(3)$ is the rotation matrix of $\mathcal{F}_i(s)$ with respect to (wrt) $\mathcal{F}_0$ and $\mathbf{p}_i \in \mathbb{R}^3$ is the position of $S_i(s)$ in $\mathcal{F}_0$. Hence, the configuration space of each beam, considered as floating, i.e. disconnected from other bodies, is naturally defined as the field of homogeneous transformations of its cross-sectional frames wrt the inertial frame $\boldsymbol{g}_i$. The space / time variations of $\boldsymbol{g}_i$ are modelled by two twist field from $[0, l_i]$ to $\mathbb{R}^6$, defined by [3]:

$$\boldsymbol{\xi}_i = \begin{pmatrix} \boldsymbol{\kappa}_i \\ \boldsymbol{\gamma}_i \end{pmatrix} = (\boldsymbol{g}_i^{-1}\boldsymbol{g}_i')^\vee \ , \ \boldsymbol{\eta}_i = \begin{pmatrix} \boldsymbol{\omega}_i \\ \boldsymbol{v}_i \end{pmatrix} = (\boldsymbol{g}_i^{-1}\dot{\boldsymbol{g}}_i)^\vee, \tag{1}$$

where $(.)' = \partial(.)/\partial s$ and $\dot{(.)} = \partial(.)/\partial t$. In these expressions, $\boldsymbol{\kappa}_i$ and $\boldsymbol{\gamma}_i$ define the angular and linear space-variation rates of cross-sectional frames along the rod, while $\boldsymbol{\omega}_i$ and $\boldsymbol{v}_i$ are their angular and linear velocities, all expressed in the mobile cross-sectional frames (note that throughout the paper, all the tensors and vectors related to a rigid body are expressed in its mobile frame). Rearranging (1) provides the s-ODE $\boldsymbol{g}_i' = \boldsymbol{g}_i\hat{\boldsymbol{\xi}}_i$, which once supplemented with initial conditions $\boldsymbol{g}_i(0) = \boldsymbol{g}_{i0}$, allows the field $\boldsymbol{g}_i$, to be reconstructed. Therefore, one can parameterize the Cosserat beam configuration by $(\boldsymbol{g}_{i0}, \boldsymbol{\xi}_i)$. Based on this parametrization, a key relation, given in [61],

[2]Following [59], we prefer to replace the word kinetostatic by the word kinemato-static: Indeed, the former is an assembly of the words kinetics and statics, and is not related with our present interest in kinematics, i.e. with the study of the motion.

[3]Note that any $\boldsymbol{g}_i$ depends on both s and t but in different ways. In Cosserat theory, s playing the role of a continuous index (a label), $\boldsymbol{g}_i$ depends on s in an explicit way. On the other hand, it depends on time in an implicit way, since its time-evolution is not imposed but governed by the static balance of forces. This explains why we note $\boldsymbol{g}_i(s)$ and not $\boldsymbol{g}_i(s,t)$.

relates the twist field $\boldsymbol{\eta}_i$ to the strain field time-derivative $\dot{\boldsymbol{\xi}}_i$ by:

$$\boldsymbol{\eta}_i(s) = \mathbf{Ad}^{-1}_{\boldsymbol{g}_i(s)} \left(\mathbf{Ad}_{\boldsymbol{g}_{i0}} \boldsymbol{\eta}_{i0} + \int_0^s \mathbf{Ad}_{\boldsymbol{g}_i(x)} \dot{\boldsymbol{\xi}}_i \, dx \right) \tag{2}$$

where we note $\boldsymbol{\eta}_{i0} = \boldsymbol{\eta}_i(0)$, while $\mathbf{Ad}_{\boldsymbol{g}}$, is a (6×6) matrix allowing a twist to be transported from a frame to another one, both frames being related by the transformation $\boldsymbol{g}$ (see Appendix A).

B. Geometrically-exact strain modes reduction

The beams considered in this paper are long and thin, thus we will neglect the shear and stretch, and use the Cosserat sub-model of inextensible Kirchhoff rods. The rods being assumed to be straight when at rest, this can be done by imposing [57]:

$$\boldsymbol{\xi}_i = \begin{pmatrix} \boldsymbol{\kappa}_i \\ \boldsymbol{\gamma}_i \end{pmatrix} = \begin{pmatrix} \boldsymbol{\Phi}_i(s) \mathbf{q}_{\mathrm{e}i} \\ \mathbf{e}_1 \end{pmatrix} \tag{3}$$

where $\mathbf{e}_1 = [1\,0\,0]^T$ and $\mathbf{q}_{\mathrm{e}i}$ is a set of generalized coordinates defined as the coefficients of the curvature field components in a basis of strain modes, i.e. we have:

$$\boldsymbol{\Phi}_i = \mathrm{diag}(\boldsymbol{\Phi}^{m_1}_{i1}, \boldsymbol{\Phi}^{m_2}_{i2}, \boldsymbol{\Phi}^{m_3}_{i3}) \tag{4}$$

with $\boldsymbol{\Phi}^{m_j}_{ij}$ a shape function row-matrix of m_j modes. For instance, one can use a simple basis of monomials $\boldsymbol{\Phi}^{m_j}_{ij} = [1 \; s \; s^2 \; \ldots \; s^{m_j - 1}]$. However, other choices motivated by numerical reasons can be adopted (e.g. orthogonal Legendre or Chebyshev polynomials, splines...). In the following of the paper, we decided to use Legendre polynomials. At the end, the vector $\mathbf{q}_{\mathrm{e}i}$ is made of $m_i = m_1 + m_2 + m_3$ components. With this further reduction, the configuration space of our n Cosserat beams is defined by the set of all possible $(\boldsymbol{g}_{i0}, \mathbf{q}_{ei})$s. As a first illustration of this reduction, note that introducing this reduced kinematics into (2) provides the reduced kinematic relation on the strain basis:

$$\boldsymbol{\eta}_i(s) = \mathbf{J}_{i1}(s) \boldsymbol{\eta}_{i0} + \mathbf{J}_{i2}(s) \dot{\mathbf{q}}_{ei}, \tag{5}$$

with $\mathbf{J}_{i1}(s) = \mathbf{Ad}^{-1}_{\boldsymbol{g}_i(s)} \mathbf{Ad}_{\boldsymbol{g}_{i0}}$ is an (6×6) matrix and $\mathbf{J}_{i2}(s) = \mathbf{Ad}^{-1}_{\boldsymbol{g}_i(s)} \mathbf{J}_{i3}(s)$, $\mathbf{J}_{i3}(s) = \int_0^s \mathbf{Ad}_{\boldsymbol{g}_i(x)} \mathbf{B} \boldsymbol{\Phi}_i \, dx$, $\mathbf{J}_{i2}$, and $\mathbf{J}_{i3}$ being $(6 \times m_i)$ matrices. It should be mentioned that in the general three-dimensional case, the expression of the matrix $\mathbf{J}_{i3}(s)$ (and thus $\mathbf{J}_{i2}(s)$) cannot be explicitly obtained, but can be still computed using standard numerical integration algorithms.

C. Reduced model of the restoring forces

In the small strains assumption, the curvature-twist field $\boldsymbol{\kappa}_i$ can be related to the stress-couple $\boldsymbol{c}_i$ by the linear constitutive relation:

$$\boldsymbol{c}_i = \mathbf{H}_a \boldsymbol{\kappa}_i \tag{6}$$

where $\mathbf{H}_a = \mathrm{diag}(G\,J_1,\, E\,J_2,\, E\,J_3)$ in which E and G are the Young's and shear moduli of the material, respectively, and J_1, J_2 and J_3 are the second moments of area (around the axes of the local frame) of the cross-sections, respectively. Introducing (3) to (4) into the definition of the virtual power of internal forces (stress), and simplifying, we get its (reduced) expression:

$$\mathcal{P}^{\star}_{int,i} = \int_0^{l_i} \boldsymbol{c}_i^T \, \dot{\boldsymbol{\kappa}}^{\star}_i \, ds = \dot{\mathbf{q}}^{\star T}_{\mathrm{e}i} \; \mathbf{K}_{\mathrm{e}i} \; \mathbf{q}_{\mathrm{e}i} \tag{7}$$

where the symbol "$\star$" indicates a virtual velocity or power, while $\mathbf{K}_{\mathrm{e}i} = \int_0^{l_i} \boldsymbol{\Phi}_i^T \, \mathbf{H}_a \, \boldsymbol{\Phi}_i \, ds$ is a constant matrix of generalized stiffness.

D. Reduced model of the external forces

Here we consider a deformable rod subject to two external wrenches exerted at each of its extremities ($\mathbf{w}_{i0} = [\mathbf{m}^T_{i0} \; \mathbf{f}^T_{i0}]^T$ at $s = 0$ and $\mathbf{w}_{il} = [\mathbf{m}^T_{il} \; \mathbf{f}^T_{il}]^T$ at $s = l_i$, where $\mathbf{f}_{i0}$ ($\mathbf{f}_{il}$, resp.) and $\mathbf{m}_{i0}$ ($\mathbf{m}_{il}$, resp.) are the force and moment of the wrench $\mathbf{w}_{i0}$ ($\mathbf{w}_{il}$, resp.)), and to a density of pose-dependent external wrench $\bar{\mathbf{w}}$ distributed along it. In these loading conditions, the virtual power of external forces is given by:

$$\mathcal{P}^{\star}_{ext,i} = \int_0^{l_i} \bar{\mathbf{w}}_i^T \boldsymbol{\eta}^{\star}_i \, ds + \mathbf{w}^T_{i0} \boldsymbol{\eta}^{\star}_{i0} + \mathbf{w}^T_{il} \boldsymbol{\eta}^{\star}_{il}, \tag{8}$$

where $\boldsymbol{\eta}_{il} = \boldsymbol{\eta}_i(l_i)$. Now, introducing the virtualization of (5), into this expression, and simplifying, allows expressing the virtual power of external forces as:

$$\mathcal{P}^{\star}_{ext,i} = \begin{bmatrix} \boldsymbol{\eta}^{\star}_{i0} \\ \dot{\mathbf{q}}^{\star}_{\mathrm{e}i} \end{bmatrix}^T \left(\int_0^{l_i} \mathbf{J}_i^T(s) \bar{\mathbf{w}}_i \, ds + \begin{bmatrix} \mathbf{w}_{i0} \\ \mathbf{0} \end{bmatrix} + \mathbf{J}_i^T(l_i) \mathbf{w}_{il} \right) \tag{9}$$

with $\mathbf{J}_i(s) = \begin{bmatrix} \mathbf{J}_{i1}(s) & \mathbf{J}_{i2}(s) \end{bmatrix}$.

E. Reduced static balance of a single rod

Finally, stating that the balance of virtual works $\mathcal{P}^{\star}_{ext,i} = -\mathcal{P}^{\star}_{int,i}$ holds for any value of $\dot{\mathbf{q}}^{\star}_{\mathrm{e}i}$ and $\boldsymbol{\eta}^{\star}_{i0}$, provides, with the expressions (7) and (9), the static balance of each beam in its reduced configuration space $\mathcal{C}^3_i$:

$$\begin{bmatrix} \mathcal{W}_i \\ \mathcal{Q}_{\mathrm{e}i} \end{bmatrix} = \begin{bmatrix} \mathbf{w}_{i0} + \bar{\mathcal{W}}_i + \mathbf{J}^T_{i1}(l_i) \mathbf{w}_{il} \\ \mathbf{K}_{\mathrm{e}i} \mathbf{q}_{\mathrm{e}i} + \mathcal{Q}_{\bar{\mathrm{w}}i} + \mathbf{J}^T_{i2}(l_i) \mathbf{w}_{il} \end{bmatrix} = \mathbf{0} \tag{10}$$

where $\bar{\mathcal{W}}_i = \int_0^{l_i} \mathbf{J}^T_{i1}(s) \bar{\mathbf{w}}_i \, ds$, $\mathcal{Q}_{\bar{\mathrm{w}}i} = \int_0^{l_i} \mathbf{J}^T_{i2}(s) \bar{\mathbf{w}}_i \, ds$, while $\mathcal{W}_i \in \mathbb{R}^6$, $\mathcal{Q}_{\mathrm{e}i} \in \mathbb{R}^{m_i}$.

F. Practical implementation of the equations

We now focus our interest on the numerical computation of (10). This computation needs to integrate the s-ODE $\boldsymbol{g}'_i = \boldsymbol{g}_i \hat{\boldsymbol{\xi}}_i$ while preserving the orthogonality of $\mathbf{R}_i$. This could be achieved by quadrature, using Magnus expansion of controlled order [62]. Here we circumvent the difficulty by parameterizing the beam rotation matrices $\mathbf{R}_i(s)$ with unit quaternions $\mathbf{h}_i(s) = h_{i0}(s) + h_{i1}(s)\mathbf{i} + h_{i2}(s)\mathbf{j} + h_{i3}(s)\mathbf{k}$, with $\mathbf{h}_i^T \mathbf{h}_i = 1$ and we have $\mathbf{R}_i(s) = \mathbf{R}_i(\mathbf{h}_i(s))$ and $\mathbf{h}'_i = \mathbf{A}(\boldsymbol{\kappa}_i)\mathbf{h}_i/2$ with $\mathbf{A}(\boldsymbol{\kappa}_i)$ a standard operator reminded in [57]. As a result, the different vectors and matrices of (10)

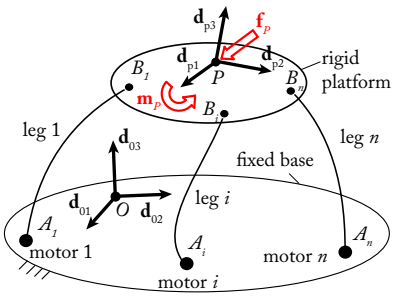


Fig. 2. A general continuum parallel robot

can be numerically computed for each leg, by applying a standard ODE integrator to:

$$\frac{\partial}{\partial s}\begin{bmatrix}\mathbf{h}_i\\ \mathbf{p}_i\\ \mathbf{J}_{i3}\\ \mathcal{W}_i\\ \mathcal{Q}_{\bar{\mathbf{w}}i}\end{bmatrix}=\begin{bmatrix}\mathbf{A}(\boldsymbol{\kappa}_i)\mathbf{h}_i/2\\ \mathbf{R}_i(\mathbf{h}_i)\boldsymbol{\gamma}_i\\ \mathbf{Ad}_{\boldsymbol{g}_i(s)}\mathbf{B}\boldsymbol{\Phi}_i\\ \mathbf{Ad}^T_{\boldsymbol{g}_{i0}}\mathbf{Ad}^{-T}_{\boldsymbol{g}_i(s)}\bar{\mathbf{w}}_i\\ \mathbf{J}^T_{i3}\mathbf{Ad}^{-T}_{\boldsymbol{g}_i(s)}\bar{\mathbf{w}}_i\end{bmatrix} \tag{11}$$

with, as inputs considered to be known: $\mathbf{q}_{ei}$, $\mathbf{h}_i(0)$ and $\mathbf{p}_i(0)$, the initial values of the other variables being null. It should be noted that, in [57], two successive forward and backward integrations of the Cosserat model with assumed modes are used in order to compute the above generalized external forces. Here, we have preferred a Jacobian formulation of the problem, which leads to the same numerical results in a single pass. Next section deals with the computation of the geometrico-static model of parallel robots.

III. GEOMETRY AND KINEMATIC DESCRIPTION OF THE CONTINUUM PARALLEL ROBOT

In this section, we first describe the general class of continuum parallel robots that we are going to analyze, as it is proposed in [21], [29]. Then, we provide the general expressions for its geometry and kinematic constraints.

A. Description of the continuum parallel robot

Let us describe the generic continuum parallel robot architecture that we consider in this paper (Fig. 2): it is a robot which is made of n slender flexible beams (called legs). An extremity of each rod is connected to a motor at one end (points A_i, $i = 1, ..., n$), the other extremity to a rigid moving platform via a joint (at points B_i, $i = 1, ..., n$), which is either a passive revolute joint, a passive spherical joint or a fixed joint, as proposed in [21], [29]. The legs can be either of constant length and connected at points A_i to an active revolute or prismatic joint as it was done in [29], or of variable length, i.e. acting like a soft cylinder fixed on the ground at point A_i as proposed in [21]. Other types of legs could be considered, by modifying the equations of the constraints and the Jacobian matrices associated to the motions of the motors. The robot moving platform, on which is located the end-effector, is considered to be rigid and endowed with the frame $\mathcal{F}_p : (P, \mathbf{d}_{\mathrm{p}1}, \mathbf{d}_{\mathrm{p}2}, \mathbf{d}_{\mathrm{p}3})$ attached it, with P the platform center of mass. Its configuration is parameterized by $\boldsymbol{g}_{\mathrm{p}} = (\mathbf{R}_{\mathrm{p}}, \mathbf{p}_{\mathrm{p}}) \in SE(3)$where $\mathbf{R}_{\mathrm{p}}$ is the rotation matrix of $\mathcal{F}_p$ with respect to $\mathcal{F}_0$, with also $\boldsymbol{\eta}_{\mathrm{p}} = (\boldsymbol{g}_{\mathrm{p}}^{-1}\dot{\boldsymbol{g}}_{\mathrm{p}})^{\vee} = [\boldsymbol{\omega}_{\mathrm{p}}^T \ \mathbf{v}_{\mathrm{p}}^T]^T \in se(3) \cong \mathbb{R}^6$, the platform twist (in the platform frame), $\boldsymbol{\omega}_{\mathrm{p}}$ and $\mathbf{v}_{\mathrm{p}}$ being the platform rotational and linear velocities, respectively. Thus, using the previous parametrization of legs, the configuration of our continuum parallel robot can first be parameterized by $\boldsymbol{g}_{\mathrm{p}}$ and the set of all the $(\boldsymbol{g}_{0\mathrm{i}}, \boldsymbol{q}_{\mathrm{ei}})$s. Note that in this definition of the configuration space, all the beams and the platform are disconnected from each other. In the next section, such a connection is performed. This concerns the connection of each leg with the base and with the platform, the first being ensured with a minimal set of coordinates, the second with a set of constraints and Lagrange multipliers.

B. Geometry and kinematics constraints of the CPR

Each leg of the CPR can be seen as a continuum beam which is subject to two types of geometry and kinematic constraints: platform and motor constraints.

1) Motor constraints: For the robots studied in this paper, motors are placed at one extremity of the beam (point A_i at $s = 0$). Other types of motors, with distributed actuation or soft cylinders (i.e. legs with varying length), could be also taken into account by using the formalism of [57]. However, we restrict our approach to the case of rods of constant length for reasons of brevity. The position of the motor i is denoted as q_{ai}, and all motor variables are grouped in the vector $\mathbf{q}_{\mathrm{a}} \in \mathbb{R}^n$. When the motor i is mounted at the proximal end of the beam i (at $s = 0$), the pose $\boldsymbol{g}_{i0} = (\mathbf{R}_i, \mathbf{p}_i)(0)$ of the proximal frame $\mathcal{F}_i(0)$, whose position and/or orientation may vary depending on the motor displacement, is a function of q_{ai}, i.e.:

$$\boldsymbol{g}_{i0} - \mathcal{G}_i(q_{ai}) = \mathbf{0} \tag{12}$$

thus leading to the generic kinematic constraints:

$$\boldsymbol{\eta}_{i0} - \mathbf{J}_{\mathrm{a}i}\dot{q}_{ai} = 0, \tag{13}$$

where $\mathbf{J}_{\mathrm{a}i} = [(1-\sigma)\mathbf{a}_i^T \ \sigma\mathbf{a}_i^T]^T$ with $\mathbf{a}_i$ the direction of the motor axis, and $\sigma = 0$ if the actuated joint is revolute, $\sigma = 1$ if it is a prismatic one. Invoking duality between efforts and velocities through virtual powers, we also have:

$$\tau_{ai}\dot{q}^{\star}_{ai} = \mathbf{w}^T_{i0}\boldsymbol{\eta}^{\star}_{i0} \text{ and thus } \tau_{ai} - \mathbf{J}^T_{\mathrm{a}i}\mathbf{w}_{i0} = 0, \tag{14}$$

where τ_{ai} is the motor force or torque. Note that (12), (13) and (14) define a reduction of the configuration space of the beams, which are now connected to the base. Formally, using these relations for modelling our CPR means that its configurations are now parameterized by $\boldsymbol{g}_{\mathrm{p}}$ and the set of all the $(\boldsymbol{q}_{\mathrm{ai}}, \boldsymbol{q}_{\mathrm{ei}})$s. Finally, in all the subsequent developments, we use this later definition of the configuration space of a CPR. In particular, the connection of the legs with the platform does not require any further reduction, since it leads to using Lagrange multipliers in the model.

2) Platform constraints: The platform constrains the extremity of each beam to be attached on it through a passive joint located at point B_i. Several types of joints, leading to different constraints, can be considered, as spherical joints, fixed joints or also revolute or cardan joints. For reasons of concision, we focus only here on spherical and fixed joints. The equations for other types of joints can be derived by using the results of [21], [33]. Thus,

- **For fixed joints:** the frame $\mathcal{F}_i(l_i)$ attached at the end of the beam i, and parameterized by $\boldsymbol{g}_{il} = (\mathbf{R}_i, \mathbf{p}_i)(l_i) = (\mathbf{R}_{il}, \mathbf{p}_{il})$, is coincident with the frame $\mathcal{F}_{pB_i} : (B_i, \mathbf{d}_{\mathrm{p}1}, \mathbf{d}_{\mathrm{p}2}, \mathbf{d}_{\mathrm{p}3})$ parameterized by $\boldsymbol{g}_{\mathrm{p}B_i} = (\mathbf{R}_{\mathrm{p}}, \mathbf{p}_{B_i})$ to within a constant transformation $\boldsymbol{g}_{ci} = (\mathbf{R}_{ci}, \mathbf{0})$, $\mathbf{R}_{ci} = \mathbf{R}_{\mathrm{p}}^T\mathbf{R}_{il}$ being a constant rotation matrix. Note that $\mathbf{p}_{B_i} = \mathbf{p}_{\mathrm{p}} + \mathbf{R}_{\mathrm{p}}\mathbf{b}_i$, vector $\mathbf{b}_i$ being a constant in $\mathcal{F}_p$. In other words, we have the geometric constraints:

$$\mathbf{p}_{\mathrm{p}} + \mathbf{R}_{\mathrm{p}}\mathbf{b}_i - \mathbf{p}_{il} = \mathbf{0} \tag{15}$$

$$(\mathbf{R}_p\mathbf{R}_{ci}\mathbf{R}_{il}^T - \mathbf{R}_{il}\mathbf{R}_{ci}^T\mathbf{R}_p^T)^\vee = \mathbf{0} \tag{16}$$

  also written as $\boldsymbol{\Psi}_i = \mathbf{0}$. Time-differentiating these geometric constraints, and simplifying the resulting expressions (or alternatively using standard twist kinematics), provides the kinematics constraints associated to (15,16):

$$\mathbf{T}_i\boldsymbol{\eta}_{\mathrm{p}} - \boldsymbol{\eta}_{il} = \mathbf{0} \tag{17}$$

  where

$$\mathbf{T}_i = \mathbf{Ad}_{\boldsymbol{g}_{il}^{-1}\boldsymbol{g}_p} = \begin{bmatrix} \mathbf{R}_{ci}^T & \mathbf{0}_{3\times3} \\ \mathbf{R}_{ci}^T\hat{\mathbf{b}}_i & \mathbf{R}_{ci}^T \end{bmatrix} \tag{18}$$

  Using (5) in $s = l_i$, and (13), (17) can be rewritten as:

$$\mathbf{T}_i\boldsymbol{\eta}_{\mathrm{p}} - \mathbf{J}_{i1}\mathbf{J}_{\mathrm{a}i}\dot{q}_{ai} - \mathbf{J}_{i2}\dot{\mathbf{q}}_{\mathrm{e}i} = \mathbf{0} \tag{19}$$

- **For spherical joints:** for such joints, there are only constraints on the positions, not on the orientations. Then, for the geometry, only constraint (15) holds while for kinematics, the constraint (17) should be rewritten as:

$$\begin{bmatrix}\mathbf{0}_{3\times3} & \mathbf{1}_{3\times3}\end{bmatrix}(\mathbf{T}_i\boldsymbol{\eta}_{\mathrm{p}} - \boldsymbol{\eta}_{il}) = \mathbf{0} \tag{20}$$

  Again, using (5), and (13), this constraint can be detailed as:

$$\begin{bmatrix}\mathbf{0}_{3\times3} & \mathbf{1}_{3\times3}\end{bmatrix}(\mathbf{T}_i\boldsymbol{\eta}_{\mathrm{p}} - \mathbf{J}_{i1}\mathbf{J}_{\mathrm{a}i}\dot{q}_{ai} - \mathbf{J}_{i2}\dot{\mathbf{q}}_{\mathrm{e}i}) = \mathbf{0} \tag{21}$$

In what follows, we will use generic notations for characterizing the platform constraints:

- for the geometry:

$$\boldsymbol{\Psi}_i(q_{ai}, \mathbf{q}_{ei}, \boldsymbol{g}_{\mathrm{p}}) = \mathbf{0} \tag{22}$$

- for the kinematics:

$$\dot{\boldsymbol{\Psi}}_i = \begin{bmatrix}\mathbf{J}_{\mathrm{q}a_i} & \mathbf{J}_{\mathrm{q}e_i} & \mathbf{J}_{\mathrm{p}_i}\end{bmatrix}\begin{bmatrix}\dot{q}_{a_i} \\ \dot{\mathbf{q}}_{ei} \\ \boldsymbol{\eta}_{\mathrm{p}}\end{bmatrix} = \mathbf{0} \tag{23}$$

  where $\mathbf{J}_{\mathrm{p}_i} = \mathbf{C}_i\mathbf{T}_i$, $\mathbf{J}_{\mathrm{q}a_i} = -\mathbf{C}_i\mathbf{J}_{i1}\mathbf{J}_{\mathrm{a}i}$, $\mathbf{J}_{\mathrm{q}e_i} = -\mathbf{C}_i\mathbf{J}_{i2}$, with $\mathbf{C}_i = \mathbf{1}_{6\times6}$ in the case of a fixed joint, $\mathbf{C}_i = \begin{bmatrix}\mathbf{0}_{3\times3} & \mathbf{1}_{3\times3}\end{bmatrix}$ in the case of a spherical joint.

3) Final form of the leg constraints: Finally, when stacking all constraints for the n robot legs:

- The total geometric constraints are defined by all relations (22), for $i = 1, \ldots, n$:

$$\boldsymbol{\Psi} = [\boldsymbol{\Psi}_1^T \ldots \boldsymbol{\Psi}_n^T]^T \tag{24}$$

  In what follows, n_Ψ denotes the number of components in $\boldsymbol{\Psi}$, and $\boldsymbol{\Psi} \in \mathbb{R}^{n_\Psi}$.
- The total kinematic constraints are defined by all the relations (23), for $i = 1, \ldots, n$:

$$\dot{\boldsymbol{\Psi}} = \mathbf{J}\,\mathbf{v} = \mathbf{0} \tag{25}$$

  where $\mathbf{v} = [\dot{\mathbf{q}}_a^T\ \dot{\mathbf{q}}_{\mathrm{e}}^T\ \boldsymbol{\eta}_{\mathrm{p}}^T]^T$, $\mathbf{q}_a = [q_{a1} \ldots q_{an}]^T \in \mathbb{R}^n$, $\mathbf{q}_{\mathrm{e}} = [\mathbf{q}_{\mathrm{e}1}^T \ldots \mathbf{q}_{\mathrm{e}n}^T]^T \in \mathbb{R}^{n_e}$, with $n_e = \sum_{i=1}^n m_i$, $\mathbf{J} = [\mathbf{J}_{\mathrm{qa}}\ \mathbf{J}_{\mathrm{qe}}\ \mathbf{J}_{\mathrm{p}}]$, with $\mathbf{J}_{\mathrm{p}} = [\mathbf{J}_{\mathrm{p}_1}^T \ \ldots\ \mathbf{J}_{\mathrm{p}_n}^T]^T \in \mathbb{R}^{n_\Psi\times6}$, and $\mathbf{J}_{\mathrm{qa}}$ and $\mathbf{J}_{\mathrm{qe}}$ two block-diagonal matrices defined by $\mathbf{J}_{\mathrm{qa}} = \mathrm{diag}(\mathbf{J}_{\mathrm{qa}_1}, \ldots, \mathbf{J}_{\mathrm{qa}_n}) \in \mathbb{R}^{n_\Psi\times n}$, $\mathbf{J}_{\mathrm{qe}} = \mathrm{diag}(\mathbf{J}_{\mathrm{qe}_1}, \ldots, \mathbf{J}_{\mathrm{qe}_n}) \in \mathbb{R}^{n_\Psi\times n_e}$. Thus, $\mathbf{J} \in \mathbb{R}^{n_\Psi\times(n+m)}$ where $m = n_e + 6$.

Note that (24) and (25) defines a geometric and kinematic model of the CPR on its configuration space.

IV. GEOMETRICO-STATIC MODEL OF THE CONTINUUM PARALLEL ROBOT

In this section, we provide the expressions for the continuum parallel robots geometrico-static model that will relate:

- the n variables in $\mathbf{q}_{\mathrm{a}}$,
- the n values of the motor generalized forces $\boldsymbol{\tau}_{\mathrm{a}} = [\tau_{a1} \ldots \tau_{an}]^T$,
- the n_e variables in $\mathbf{q}_{\mathrm{e}}$,
- the six independent components of the group transformation $\boldsymbol{g}_{\mathrm{p}}$,
- the n_Ψ components of a set of Lagrange multipliers $\boldsymbol{\lambda}$,

to the robot external loading. Furthermore, we discuss the algorithmic implementation of the equations.

A. Geometrico-static model

According to our definition of the configuration space of a CPR, we need to set the static balance of the platform in $SE(3)$ (indeed, the space of wrenches $se(3)^\star$), and those of legs connected to the base, in the space of their motor and strain coordinates. These equations are related together through a set of reaction wrenches whose components define some Lagrange multipliers in charge of forcing the constraints imposed by the connection of the legs with the platform. Let us now consider the static balance of the platform: each leg exerts on it a wrench $\mathbf{w}_{il} = [\mathbf{m}_{il}^T\ \mathbf{f}_{il}^T]^T$ through the joint at point B_i, which balances an external wrench $\mathbf{w}_{\mathrm{p}} = [\mathbf{m}_{\mathrm{p}}^T\ \mathbf{f}_{\mathrm{p}}^T]^T$, which is expressed in the platform frame ($\mathbf{f}_{\mathrm{p}}$ is its force, $\mathbf{m}_{\mathrm{p}}$ its moment), and includes the gravity effects. As a result, the force and moment static balances are, at point P, and in the inertial basis:

$$\mathbf{R}_{\mathrm{p}}\mathbf{f}_{\mathrm{p}} - \sum_{i=1}^{n}\mathbf{R}_i\mathbf{f}_{il} = \mathbf{0} \tag{26}$$

$$\mathbf{R}_{\mathrm{p}}\mathbf{m}_{\mathrm{p}} - \sum_{i=1}^{n}\mathbf{R}_i\mathbf{m}_{il} - \sum_{i=1}^{n}\mathbf{R}_{\mathrm{p}}\mathbf{b}_i \times \mathbf{R}_i\mathbf{f}_{il} = \mathbf{0} \tag{27}$$

or also, in terms of wrenches:

$$\mathbf{w}_{\mathrm{p}} - \sum_{i=1}^{n} \mathbf{T}_i^T \mathbf{w}_{il} = \mathbf{0} \tag{28}$$

where $\mathbf{T}_i$ is defined as in (18). Assuming the joints to be ideal, the components of $\mathbf{w}_{il}$ along their allowed degrees of freedom are zero, while the others define the vector of Lagrange multipliers $\boldsymbol{\lambda}_i$ according to the relation:

$$\mathbf{w}_{il} = \mathbf{C}_i^T \boldsymbol{\lambda}_i \tag{29}$$

where $\mathbf{C}_i$ is a selection matrix defined below (23). Introducing (29) into (28) provides the expected platform static balance of wrenches:

$$\mathbf{w}_{\mathrm{p}} - \sum_{i=1}^{n} \mathbf{J}_{\mathrm{p}i}^T \boldsymbol{\lambda}_i = \mathbf{0} \tag{30}$$

where we used the notations below (23). To get the static balance of legs, we introduce (29) into (10), that we project onto the legs configuration space by pre-multiplying the first of their rows by $\mathbf{J}_{\mathrm{a}i}^T$. This provides the expected static balances of the legs:

$$\begin{bmatrix} \mathcal{Q}_{\mathrm{a}i} \\ \mathcal{Q}_{\mathrm{e}i} \end{bmatrix} = \begin{bmatrix} \tau_{ai} + \mathbf{J}_{\mathrm{a}i}^T \bar{\mathcal{W}}_i - \mathbf{J}_{\mathrm{qa}_i}^T \boldsymbol{\lambda}_i \\ \mathbf{K}_{\mathrm{e}i} \mathbf{q}_{\mathrm{e}i} + \mathcal{Q}_{\bar{\mathrm{w}}i} - \mathbf{J}_{\mathrm{qe}_i}^T \boldsymbol{\lambda}_i \end{bmatrix} = \mathbf{0} \tag{31}$$

where we used the notations introduced in (23), while remark that $\mathcal{Q}_{\mathrm{a}i} \in \mathbb{R}$. Now, stacking (24) and (30) with all equations (31) for $i = 1, \ldots, n$, provides **the expression of the geometrico-static model of a *CPR***:

$$\left\{ \begin{array}{rcl} \mathcal{E} = \mathcal{H} - \mathbf{J}^T \boldsymbol{\lambda} & = & 0 \\ \boldsymbol{\Psi} & = & 0 \end{array} \right. \tag{32}$$

where $\mathbf{J}$ is defined by (25), $\mathcal{H} = [\mathcal{H}_{\mathrm{a}}^T \ \mathcal{H}_{\mathrm{e}}^T \ \mathcal{H}_{\mathrm{p}}^T]^T \in \mathbb{R}^{n+n_e+6}$ in which:

- $\mathcal{H}_{\mathrm{a}} = [\mathcal{H}_{\mathrm{a}1}^T \ldots \mathcal{H}_{\mathrm{a}n}^T]^T \in \mathbb{R}^n$ with
$$\mathcal{H}_{\mathrm{a}i} = \tau_{ai} + \mathbf{J}_{\mathrm{a}i}^T \bar{\mathcal{W}}_i \in \mathbb{R} \tag{33}$$
$\bar{\mathcal{W}}_i$ being defined in (10),
- $\mathcal{H}_{\mathrm{e}} = [\mathcal{H}_{\mathrm{e}1}^T \ldots \mathcal{H}_{\mathrm{e}n}^T]^T \in \mathbb{R}^{n_e}$ with
$$\mathcal{H}_{\mathrm{e}i} = \mathbf{K}_{\mathrm{e}i} \mathbf{q}_{\mathrm{e}i} + \mathcal{Q}_{\bar{\mathrm{w}}i} \in \mathbb{R}^{m_i} \tag{34}$$
$\mathcal{Q}_{\bar{\mathrm{w}}i}$ being defined in (10),
- $\mathcal{H}_{\mathrm{p}} = \mathbf{w}_{\mathrm{p}} \in \mathbb{R}^6$,
- $\boldsymbol{\lambda} = [\boldsymbol{\lambda}_1^T \ldots \boldsymbol{\lambda}_n^T]^T \in \mathbb{R}^{n_\Psi}$ can be seen as a vector of Lagrange multipliers, that gathers all the independent components of the wrenches $\mathbf{w}_{il}$.

The number of equations in (32) is thus equal to $n+m+n_\Psi$ (remind that $m = n_e + 6$), since we have:

- as many equations in $\mathcal{E}$ as the number of variables in $[\dot{\mathbf{q}}_a^T \ \dot{\mathbf{q}}_e^T \ \boldsymbol{\eta}_{\mathrm{p}}^T]^T$, i.e. there are $n + n_e + 6$ equations in $\mathcal{E}$.
- n_Ψ equations in $\boldsymbol{\Psi}$.

The number of unknowns to be found is equal to $2n+m+n_\Psi$, since we have:

- n variables in $\mathbf{q}_a$ and n_e variables in $\mathbf{q}_e$,
- 6 independent variables in $\boldsymbol{g}_{\mathrm{p}}$,
- n variables in $\boldsymbol{\tau}_a$ and n_Ψ variables in $\boldsymbol{\lambda}$.

Thus, there is a total of $n+m+n_\Psi$ equations for $2n+m+n_\Psi$ unknowns in (32). Finding solutions to (32) is the goal of the next section.

B. Forward and inverse geometrico-static problems

The geometrico-static model (32) is a system of $n + m + n_\Psi$ equations and $2n + m + n_\Psi$ unknowns $(\mathbf{q}, \boldsymbol{\tau}_{\mathrm{a}}, \boldsymbol{\lambda}) = (\mathbf{q}_{\mathrm{a}}, \mathbf{q}_{\mathrm{p}}, \mathbf{q}_{\mathrm{u}}, \boldsymbol{\tau}_{\mathrm{a}}, \boldsymbol{\lambda})$, where $\mathbf{q}_{\mathrm{p}}$ denotes a vector of n controlled outputs generally chosen among the parameters of the platform while $\mathbf{q}_{\mathrm{u}}$ is the vector of residual uncontrolled coordinates. As a consequence, fixing n variables to some desired values, provides a square system of equations, having generically a finite number of solutions. The *forward geometrico-static problem* consists in fixing the n motor positions $\mathbf{q}_{\mathrm{a}}$ to some desired values and to compute the corresponding n controlled coordinates, the n input torques $\boldsymbol{\tau}_{\mathrm{a}}$, the m uncontrolled coordinates $\mathbf{q}_{\mathrm{pu}}$ and the n_Ψ Lagrange multipliers $\boldsymbol{\lambda}$ so that $(\mathbf{q}, \boldsymbol{\tau}_{\mathrm{a}}, \boldsymbol{\lambda})$ is solution of the implicit geometrico-static model (32). It should be mentioned that a variant of the forward geometrico-static problem consists in fixing $\boldsymbol{\tau}_{\mathrm{a}}$, instead of $\mathbf{q}_{\mathrm{a}}$. However, we will not further discuss about this alternative choice. The *inverse geometrico-static problem* consists in fixing the n controlled coordinates $\mathbf{q}_{\mathrm{p}}$ to some desired values, and to compute the corresponding n motor positions $\mathbf{q}_{\mathrm{a}}$, the n input torques $\boldsymbol{\tau}_{\mathrm{a}}$, the m uncontrolled coordinates $\mathbf{q}_{\mathrm{u}}$, and the n_Ψ Lagrange multipliers $\boldsymbol{\lambda}$, so that $(\mathbf{q}, \boldsymbol{\tau}_{\mathrm{a}}, \boldsymbol{\lambda})$ is solution of the implicit geometrico-static model (32).

In both cases, the computed configurations $\mathbf{q}_{\mathrm{a}}$, $\mathbf{q}_{\mathrm{p}}$ and $\mathbf{q}_{\mathrm{u}}$, and input efforts $\boldsymbol{\tau}_{\mathrm{a}}$, are only local extrema that must be additionally checked to be local minimizers of the potential energy (see [33]), i.e., to be actual stable configurations. The computed multipliers $\boldsymbol{\lambda}$ are useless in practice, but they actually have to be computed when solving Lagrange conditions associated to an equality constrained optimization problem. Moreover, they are necessary to assess the stability of the associated solution. The stability conditions of the computed local extrema, are detailed further in Section V-D.

C. Algorithmic implementation

For solving this system of equations, standard solvers can be used (e.g. Newton-Raphson, Levenberg-Marquardt, Trust Region algorithms [63]: those three methods are available in the *fsolve* function in Matlab). In any case, the inputs/outputs will be:

- for the forward geometric problem:
  - inputs: imposed values for $\mathbf{q}_{\mathrm{a}}$, and an initial guess for $\mathbf{q}_{\mathrm{pu}} = [\mathbf{q}_{\mathrm{p}} \, \mathbf{q}_{\mathrm{u}}]^T$, $\boldsymbol{\tau}_{\mathrm{a}}$ and $\boldsymbol{\lambda}$,
  - outputs: values for $\mathbf{q}_{\mathrm{pu}}$, $\boldsymbol{\tau}_{\mathrm{a}}$ and $\boldsymbol{\lambda}$.
- for the inverse geometric problem:
  - inputs: imposed values for $\mathbf{q}_{\mathrm{p}}$, and an initial guess for $\mathbf{q}_{\mathrm{au}} = [\mathbf{q}_{\mathrm{a}} \, \mathbf{q}_{\mathrm{u}}]^T$, $\boldsymbol{\tau}_{\mathrm{a}}$ and $\boldsymbol{\lambda}$,
  - outputs: values for $\mathbf{q}_{\mathrm{au}}$, $\boldsymbol{\tau}_{\mathrm{a}}$ and $\boldsymbol{\lambda}$.

Whatever the type of problem, the geometrico-static model (32) will be solved by following the steps detailed hereafter:

1) Provide initial values for $\mathbf{q}$, $\boldsymbol{\tau}_{\mathrm{a}}$ and $\boldsymbol{\lambda}$: the algorithm will update the values of the unknowns only, and keep constant the values that have been fixed ($\mathbf{q}_{\mathrm{a}}$ for the forward prolem, $\mathbf{q}_{\mathrm{p}}$ for the inverse problem),

2) For $i = 1, \ldots, n$, solve the system of ODEs (11) $\Rightarrow$ Obtain the values of $\bar{\mathcal{W}}_i$, $\mathcal{Q}_{\bar{\mathrm{w}}i}$, $\mathbf{p}_{il}$, $\mathbf{R}_{il}$ and $\mathbf{J}_{i2}(l_i)$; in this work, we used a standard Runge-Kutta 45 solver (function *ode45* in Matlab), but another type of solver could have been chosen (see for instance [64])
3) Compute $\mathcal{H}$, $\mathbf{\Psi}$ and $\mathbf{J}$, then compute the model equations (32),
4) Iterate with the nonlinear equation solver (i.e. change the values of the unknowns in $\mathbf{q}$, $\boldsymbol{\tau}_\mathrm{a}$ and $\boldsymbol{\lambda}$) as long as both $\mathcal{E}$ and $\mathbf{\Psi}$ are not equal to $\mathbf{0}$.

It should be mentioned here that using one of the above mentioned standard iterative solvers (e.g. Newton-Raphson, Levenberg-Marquardt, Trust Region algorithms [63]), requires to compute the Jacobian matrices of the model. By default, this can be achieved numerically by resorting to finite differences approximations. However, using their exact analytical expressions allows to considerably speed up the computational time. This second approach is here used and the corresponding expressions of the Jacobian matrices are detailed in the report [61].

Next Section deals with the computation of the kinemato-static model.

V. KINEMATO-STATIC MODEL OF THE CONTINUUM PARALLEL ROBOT

In this Section, we give the expression of the kinemato-static model, which relates

- The variations $\Delta\mathbf{q}_\mathrm{a}$, $\Delta\mathbf{q}_\mathrm{e}$, of the motor and elastic coordinates
- The variation $\Delta\mathbf{\Sigma}_\mathrm{p} = (\boldsymbol{g}_p^{-1}\Delta\boldsymbol{g}_p)^\vee \in se(3) \cong \mathbb{R}^6$ of the platform configuration (in the platform frame). Note that if this variation were performed with respect to time, it would merely coincide with the platform twist $\boldsymbol{\eta}_\mathrm{p}$,
- To the variations $\Delta\boldsymbol{\tau}_\mathrm{a}$, $\Delta\mathbf{w}_\mathrm{p}$ and $\Delta\boldsymbol{\lambda}$ of the input efforts, platform wrenches and Lagrange multipliers,

where, in these variables, the symbol $\Delta(.)$ stands for a small variation of the variable written on its right-hand side. These models are derived in order to define the singularity cases of the CPR.

We also discuss the conditions for robot stability.

A. Derivatives of the geometrico-static model

The kinemato-static model of the robot can be derived by computing the variation of the equations (32) w.r.t. all variables $\mathbf{q}_\mathrm{a}$, $\mathbf{q}_\mathrm{e}$, $\mathbf{p}_\mathrm{p}$ and $\mathbf{h}_\mathrm{p}$, w.r.t. multipliers $\boldsymbol{\lambda}$ as well, but also w.r.t. $\boldsymbol{\tau}_\mathrm{a}$ and $\mathbf{w}_\mathrm{p}$. Thus, the kinemato-static model is given by:

$$\begin{aligned}\Delta\mathcal{E} =&\mathbf{P}_\mathcal{E}\Delta\mathbf{\Sigma}_\mathrm{p} + \left[\frac{\partial\mathcal{E}}{\partial\mathbf{q}_\mathrm{a}}\right]\Delta\mathbf{q}_\mathrm{a} \\ &+ \left[\frac{\partial\mathcal{E}}{\partial\mathbf{q}_\mathrm{e}}\right]\Delta\mathbf{q}_\mathrm{e} + \left[\frac{\partial\mathcal{E}}{\partial\boldsymbol{\lambda}}\right]\Delta\boldsymbol{\lambda} + \left[\frac{\partial\mathcal{E}}{\partial\mathbf{w}}\right]\Delta\mathbf{w} = \mathbf{0}\end{aligned} \tag{35}$$

where $\Delta\mathbf{w} = [\Delta\boldsymbol{\tau}_\mathrm{a}^T\ \Delta\mathbf{w}_\mathrm{p}^T]^T \in \mathbb{R}^{6+n}$, and the kth component $\mathbf{p}_{\mathcal{E}k}$ of $\mathbf{P}_\mathcal{E}$ is given by:

$$\mathbf{p}_{\mathcal{E}k} = \begin{bmatrix}\frac{\partial\mathcal{E}_k}{\partial\mathbf{p}_\mathrm{p}}^T\mathbf{R}_\mathrm{p} & \left(\mathbf{R}_\mathrm{p}^T\frac{\partial\mathcal{E}_k}{\partial\mathbf{R}_\mathrm{p}} - \frac{\partial\mathcal{E}_k}{\partial\mathbf{R}_\mathrm{p}}^T\mathbf{R}_\mathrm{p}\right)^{\vee T}\end{bmatrix} \in \mathbb{R}^{1\times 6} \tag{36}$$

with, from [48], $\mathcal{E}_k$ the kth component of $\mathcal{E}$. Additionally,

$$\Delta\mathbf{\Psi} = \mathbf{J}_\mathrm{qa}\Delta\mathbf{q}_a + \mathbf{J}_\mathrm{qe}\Delta\mathbf{q}_\mathrm{e} + \mathbf{J}_\mathrm{p}\Delta\mathbf{\Sigma}_\mathrm{p} = \mathbf{0} \tag{37}$$

As a result, the kinemato-static model of the CPR can be written as:

$$\mathbf{W}_\mathcal{E}\Delta\mathbf{w} = \mathbf{P}_\mathcal{E}\Delta\mathbf{\Sigma}_\mathrm{p} + \mathbf{A}_\mathcal{E}\Delta\mathbf{q}_\mathrm{a} + \mathbf{E}_\mathcal{E}\Delta\mathbf{q}_\mathrm{e} + \mathbf{\Lambda}_\mathcal{E}\Delta\boldsymbol{\lambda} \tag{38}$$

$$\mathbf{0} = \mathbf{J}_\mathrm{p}\Delta\mathbf{\Sigma}_\mathrm{p} + \mathbf{J}_\mathrm{qa}\Delta\mathbf{q}_a + \mathbf{J}_\mathrm{qe}\Delta\mathbf{q}_\mathrm{e} \tag{39}$$

with:

- $\mathbf{A}_\mathcal{E} = \left[\frac{\partial\mathcal{E}}{\partial\mathbf{q}_\mathrm{a}}\right] \in \mathbb{R}^{(n+m)\times n}$,
- $\mathbf{E}_\mathcal{E} = \left[\frac{\partial\mathcal{E}}{\partial\mathbf{q}_\mathrm{e}}\right] \in \mathbb{R}^{(n+m)\times n_e}$,
- $\mathbf{\Lambda}_\mathcal{E} = \left[\frac{\partial\mathcal{E}}{\partial\boldsymbol{\lambda}}\right] = \mathbf{J}^T \in \mathbb{R}^{(n+m)\times n_\Psi}$ and

$$\mathbf{W}_\mathcal{E} = -\begin{bmatrix}\mathbf{1}_{n\times n} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{1}_{6\times 6}\end{bmatrix} \in \mathbb{R}^{(n+m)\times(n+6)}.$$

The expressions of the Jacobian matrices $\mathbf{A}_\mathcal{E}$ and $\mathbf{E}_\mathcal{E}$ can be obtained by using the approach [65]. They are provided in the report [61].

B. Kinemato-static model of the CPR

For the kinemato-static analysis, there is little interest to characterize the variations $\Delta\boldsymbol{\lambda}$ w.r.t. to the others. Therefore, it is worthy to eliminate the multipliers variations $\Delta\boldsymbol{\lambda}$ from the system of equations. Removing $\Delta\boldsymbol{\lambda}$ can be done by using the matrix $\mathbf{Z}$ spanning the nullspace of the matrix $\mathbf{\Lambda}_\mathcal{E}^T$, which satisfies $\mathbf{Z}^T\mathbf{\Lambda}_\mathcal{E} = \mathbf{0}$ by definition. Under the assumption that $\mathbf{\Lambda}_\mathcal{E}$ is full rank, the matrix $[\mathbf{Z}\,\mathbf{\Lambda}_\mathcal{E}]$ is square and nonsingular. Therefore, left-multiplying (38) by the nonsingular square matrix $[\mathbf{Z}\,\mathbf{\Lambda}_\mathcal{E}]^T$ gives rise to the following block-triangularized system equivalent to (38):

$$\mathbf{0} = \mathbf{Z}^T\left(\mathbf{P}_\mathcal{E}\Delta\mathbf{\Sigma}_\mathrm{p} + \mathbf{A}_\mathcal{E}\Delta\mathbf{q}_\mathrm{a} + \mathbf{E}_\mathcal{E}\Delta\mathbf{q}_\mathrm{e} - \mathbf{W}_\mathcal{E}\Delta\mathbf{w}\right) \tag{40}$$

$$\mathbf{0} = \mathbf{\Lambda}_\mathcal{E}^T\left(\mathbf{P}_\mathcal{E}\Delta\mathbf{\Sigma}_\mathrm{p} + \mathbf{A}_\mathcal{E}\Delta\mathbf{q}_\mathrm{a} + \mathbf{E}_\mathcal{E}\Delta\mathbf{q}_\mathrm{e} + \mathbf{\Lambda}_\mathcal{E}\Delta\boldsymbol{\lambda} - \mathbf{W}_\mathcal{E}\Delta\mathbf{w}\right). \tag{41}$$

Finally, we can gather Equations (40) and (39) to obtain the **final expression of the kinemato-static model of a *CPR***:

$$\mathbf{A}\,\Delta\mathbf{q}_\mathrm{a} + \mathbf{P}\,\Delta\mathbf{\Sigma}_\mathrm{p} + \mathbf{E}\,\Delta\mathbf{q}_\mathrm{e} = \mathbf{W}\Delta\mathbf{w}, \tag{42}$$

where

$$\begin{aligned}&\mathbf{A} = \begin{bmatrix}\mathbf{Z}^T\mathbf{A}_\mathcal{E} \\ \mathbf{J}_\mathrm{qa}\end{bmatrix} \in \mathbb{R}^{r\times n},\ \mathbf{P} = \begin{bmatrix}\mathbf{Z}^T\mathbf{P}_\mathcal{E} \\ \mathbf{J}_\mathrm{p}\end{bmatrix} \in \mathbb{R}^{r\times 6}, \\ &\mathbf{E} = \begin{bmatrix}\mathbf{Z}^T\mathbf{E}_\mathcal{E} \\ \mathbf{J}_\mathrm{qe}\end{bmatrix} \in \mathbb{R}^{r\times n_e}, \mathbf{W} = \begin{bmatrix}\mathbf{Z}^T\mathbf{W}_\mathcal{E} \\ \mathbf{0}\end{bmatrix} \in \mathbb{R}^{r\times(n+6)}\end{aligned} \tag{43}$$

where $r = n+m$ (recall that $m = n_e+6$). Remaining Eq. (41) can be then used in order to compute the variation $\Delta\boldsymbol{\lambda}$.

Note also that the same elimination process could be used to remove the variables $\Delta\mathbf{q}_\mathrm{e}$ from (42), provided that $\mathbf{E}$ is a full rank matrix.

C. Forward and inverse kinematico-static problems

Using the definition of $\Delta\mathbf{w}$ below (35), one can introduce the two matrices $\mathbf{W}_\mathrm{a}$ and $\mathbf{W}_\mathrm{p}$ and detail the right hand side of (42) as:

$$\mathbf{W}\Delta\mathbf{w} = -\mathbf{W}_\mathrm{a}\Delta\boldsymbol{\tau}_\mathrm{a} - \mathbf{W}_\mathrm{p}\Delta\mathbf{w}_\mathrm{p}. \tag{44}$$

A square system of equations is obtained by fixing $n+6$ components in the vectors $\Delta\mathbf{q}_\mathrm{a}$, $\Delta\mathbf{q}_\mathrm{e}$, $\Delta\boldsymbol{\Sigma}_\mathrm{p}$ and $\Delta\mathbf{w}_\mathrm{p}$, leading to the forward and inverse kinemato-static problems:

- *Forward kinemato-static problem*: Given $\Delta\mathbf{q}_\mathrm{a} \in \mathbb{R}^n$ and usually $\Delta\mathbf{w}_\mathrm{p} \in \mathbb{R}^6$, compute $\Delta\mathbf{q}_\mathrm{e}$, $\Delta\boldsymbol{\Sigma}_\mathrm{p}$ and $\Delta\boldsymbol{\tau}_\mathrm{a}$ by solving the linear system (42), which becomes a system of r equations and r unknowns. Provided that matrix $\begin{bmatrix}\mathbf{P} & \mathbf{E} & \mathbf{W}_\mathrm{a}\end{bmatrix}$ is nonsingular, the solution of the forward kinemato-static problem is
$$\begin{bmatrix}\Delta\boldsymbol{\Sigma}_\mathrm{p}\\ \Delta\mathbf{q}_\mathrm{e}\\ \Delta\boldsymbol{\tau}_\mathrm{a}\end{bmatrix} = -\begin{bmatrix}\mathbf{P} & \mathbf{E} & \mathbf{W}_\mathrm{a}\end{bmatrix}^{-1}\begin{bmatrix}\mathbf{A} & \mathbf{W}_\mathrm{p}\end{bmatrix}\begin{bmatrix}\Delta\mathbf{q}_\mathrm{a}\\ \Delta\mathbf{w}_\mathrm{p}\end{bmatrix}. \tag{45}$$
  It should be mentioned that instead of putting the vector $\Delta\mathbf{w}_\mathrm{p}$ as input of the forward kinemato-static problem, 6 other components of the vector $\begin{bmatrix}\Delta\boldsymbol{\Sigma}_\mathrm{p}^T & \Delta\mathbf{q}_\mathrm{e}^T & \Delta\boldsymbol{\tau}_\mathrm{a}^T\end{bmatrix}^T$ could have been chosen.
- *Inverse kinemato-static problem*: Given $\Delta\boldsymbol{\Sigma}_\mathrm{p} \in \mathbb{R}^6$ and usually $\Delta\boldsymbol{\tau}_\mathrm{a} \in \mathbb{R}^n$, compute $\Delta\mathbf{q}_\mathrm{a}$, $\Delta\mathbf{q}_\mathrm{e}$ and $\Delta\mathbf{w}_\mathrm{p}$ by solving the linear system (42), which becomes again a system of $n+m$ equations and $n+m$ unknowns. Provided that matrix $\begin{bmatrix}\mathbf{A} & \mathbf{E} & \mathbf{W}_\mathrm{p}\end{bmatrix}$ is nonsingular, the solution of the inverse kinemato-static problem is
$$\begin{bmatrix}\Delta\mathbf{q}_\mathrm{a}\\ \Delta\mathbf{q}_\mathrm{e}\\ \Delta\mathbf{w}_\mathrm{p}\end{bmatrix} = -\begin{bmatrix}\mathbf{A} & \mathbf{E} & \mathbf{W}_\mathrm{p}\end{bmatrix}^{-1}\begin{bmatrix}\mathbf{P} & \mathbf{W}_\mathrm{a}\end{bmatrix}\begin{bmatrix}\Delta\boldsymbol{\Sigma}_\mathrm{p}\\ \Delta\boldsymbol{\tau}_\mathrm{a}\end{bmatrix}. \tag{46}$$
  It should also be mentioned that instead of setting the vector $\Delta\boldsymbol{\tau}_\mathrm{a}$ as an input of the inverse kinemato-static problem, n other components of the vector $\begin{bmatrix}\Delta\mathbf{q}_\mathrm{a}^T & \Delta\mathbf{q}_\mathrm{e}^T & \Delta\mathbf{w}_\mathrm{p}^T\end{bmatrix}^T$ could have been used.

Finally, the unsolvability of the forward and inverse kinemato-static problems due to the nonregularity of the matrices to be inverted in (45) and (46) leads to the main conditions of singularities. General conditions of singularities of the kinemato-static model have been analyzed in [33].

D. Stability analysis of the CPR

For analyzing the robot static stability, let us come back to the definition of its potential energy. Here, external 3D moments being non-conservative [66], we consider that none of them is applied on the robot. As a result, its total potential energy U is provided by the expression:

$$U = \sum_{i=1}^{n}\int_0^{l_i}\left(\frac{1}{2}\boldsymbol{c}_i^T\boldsymbol{\kappa}_i - \bar{\mathbf{f}}_i^T\mathbf{p}_i\right)ds - {}^0\mathbf{f}_\mathrm{p}^T\mathbf{p}_\mathrm{p} - \boldsymbol{\tau}_\mathrm{a}^T\mathbf{q}_\mathrm{a} \tag{47}$$

where $\bar{\mathbf{f}}_i$ is a distributed external force on the leg i, ${}^0\mathbf{f}_\mathrm{p}$ is a force applied on the platform, constant in the world frame, and the expressions of $\boldsymbol{c}_i$ and $\boldsymbol{\kappa}_i$ are provided as a function of $\mathbf{q}_\mathrm{e}$ in Section II. A solution of the geometrico-static model is necessarily a configuration which minimizes the potential energy U under the geometric constraints $\boldsymbol{\Psi} = \mathbf{0}$, for a fixed value of $\boldsymbol{\tau}_\mathrm{a}$, i.e. it is a solution to the following optimization problem:

$$(\boldsymbol{g}_\mathrm{p}, \mathbf{q}_\mathrm{a}, \mathbf{q}_\mathrm{e}) = \mathrm{argmin}(U) \text{ subject to } \boldsymbol{\Psi} = \mathbf{0} \tag{48}$$

A solution to this optimization problem is also a solution to the equation:

$$\delta\mathcal{L} = 0 \tag{49}$$

where $\mathcal{L}$ is a Lagrangian function given by $\mathcal{L} = U + \boldsymbol{\Psi}^T\boldsymbol{\lambda}$ and $\delta\mathcal{L}$ is its first variation which is a function of the variations $\delta\boldsymbol{\Sigma}_\mathrm{p}$, $\delta\mathbf{q}_\mathrm{a}$, $\delta\mathbf{q}_\mathrm{e}$ and $\delta\boldsymbol{\lambda}$, i.e.

$$\delta\mathcal{L} = \mathbf{p}_\mathcal{L}^T\delta\boldsymbol{\Sigma}_\mathrm{p} + \mathbf{a}_\mathcal{L}^T\delta\mathbf{q}_\mathrm{a} + \mathbf{e}_\mathcal{L}^T\delta\mathbf{q}_\mathrm{e} + \mathbf{l}_\mathcal{L}^T\delta\boldsymbol{\lambda} = 0 \tag{50}$$

Skipping the mathematical derivations, it can be proven that $[\mathbf{a}_\mathcal{L}^T\ \mathbf{e}_\mathcal{L}^T\ \mathbf{p}_\mathcal{L}^T]^T = \mathcal{E}$ and that $\mathbf{l}_\mathcal{L} = \boldsymbol{\Psi}$, i.e. the variation (50) leads to the geometrico-static model. As a result, the right-hand side of the equations (38) and (39) defines the second variation of $\mathcal{L}$ as a function of the variations $\Delta\boldsymbol{\Sigma}_\mathrm{p}$, $\Delta\mathbf{q}_\mathrm{a}$, $\Delta\mathbf{q}_\mathrm{e}$ and $\Delta\boldsymbol{\lambda}$. As shown in [63], the stability of the configuration can thus be checked by analysing the positive-definiteness of the matrix $\mathbf{H}_r$ defined as:

$$\mathbf{H}_r = \mathbf{Z}^T\mathbf{H}\mathbf{Z} \tag{51}$$

where

$$\mathbf{H} = \begin{bmatrix}\mathbf{A}_\mathcal{E} & \mathbf{E}_\mathcal{E} & \mathbf{P}_\mathcal{E}\end{bmatrix} \tag{52}$$

all these matrices being given above. Let us now deal with some case studies.

VI. CASE STUDIES

In this section, we will model two robots (a planar robot with two degrees of controllability, and a spatial one, with two degrees of controllability) and compare the model prediction of our approach with the full continuous approach of [21].

A. The planar $\underline{R}FRF\underline{R}$ robot

In this Section, we study the continuum planar parallel robot $\underline{R}FRF\underline{R}$ robot made of two rods which has been presented in [26] (Fig. 3). It is composed of two actuated revolute ($\underline{R}$) joints, each being mounted on the ground and attached at one extremity of a flexible rod (F). Both flexible rods are connected at their extremity through a passive revolute (R) joint. The pose $\mathbf{q}_\mathrm{p}$ is the coordinates of the point P denoted by (x, y).

Parameters of the rods are as follows: They are straight at rest, with length $L = 1$ m, and circular cross-sections of radius 1 mm, their Young's and shear moduli are $E = 210$ GPa and $G = 82$ GPa, respectively, and their density is $\rho = 7800$ kg/m^3. The distance $\ell_{A_1A_2}$ between the two motors is $\ell_{A_1A_2} = 0.4$ m. No external wrench is applied. The joint at the end-effector (point P) is considered to be massless.

The robot is modeled with the planar deformation assumption used in [26], [44] and Kirchhoff internal kinematics (i.e. inextensibility and unshearability are assumed). Numerical resolution is performed with our approach taking the same

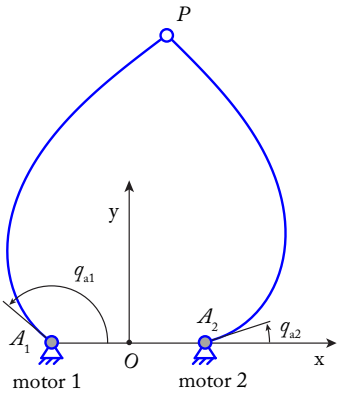


Fig. 3. Schematics of the $\underline{R}FRF\underline{R}$ robot (to scale).

number of bending modes m_j per leg (in what follows, $m_j = 3$, 4 or 5), and with the continuous approach and shooting algorithm of [21] (also restricted to the planar Kirchhoff kinematics) with Matlab. The reduced approach of the article is then compared to [21] in terms of accuracy of prediction for the geometrico-static model, computational time, and stability prediction.

It should be mentioned that, for all computations shown thereafter (for both our model and the continuous Cosserat model [21]), the solver used is the Levenberg-Marquardt algorithm encoded in the Matlab *fsolve* function, with an initial damping of 0.1.

1) Model prediction accuracy: For checking the model accuracy, let us first estimate the position of the end-effector both with our model and the model [21] along a path followed by the motors defined by:

$$\mathbf{q}_{\mathrm{a}}(t) = \frac{t}{n_c}(\mathbf{q}_{\mathrm{af}} - \mathbf{q}_{\mathrm{a0}}) + \mathbf{q}_{\mathrm{a0}} \tag{53}$$

where $\mathbf{q}_{\mathrm{a0}}$ ($\mathbf{q}_{\mathrm{af}}$, resp.) is the motor initial (final, resp.) configuration, n_c is the number of tested configuration along the path and t an integer between 0 and n_c. Here, we took $\mathbf{q}_{\mathrm{a0}} = [2.59\ 0.55]^T$ rad, $\mathbf{q}_{\mathrm{af}} = [5.76\ 3.67]^T$ rad. An example of computed configurations along this path for $n_c = 10$ is shown in Fig. 4.

The error of prediction between our model (denoted as Mod. #1) and the model [21] (denoted as Mod. #2)[4] for the end-effector position for $n_c+1 = 51$ configurations computed by feeding the forward geometrico-static models with the equation (53) is shown in Fig. 5(a), and a summary of the results is presented in Table I. As expected, the prediction error decreases with the number of modes. With five modes per leg (i.e. $\mathbf{q}_{\mathrm{e}}$ has 10 components in totality), the mean error is lower than 30 microns, which is usually much enough for a robot with legs of 1 m, when considering all disturbing unmodelled phenomena that could arise in a real experimentation.

We then compared the same robot modelled with a finite difference approach as in [44] (denoted as Mod. #3) with Mod. #2. Indeed, finite differences is a standard procedure for solving the Cosserat model, which is used in many papers

[4]Here, we consider that the numerical solution of the model [21] is the ground truth. However, even ODE solvers are prone to solution errors.

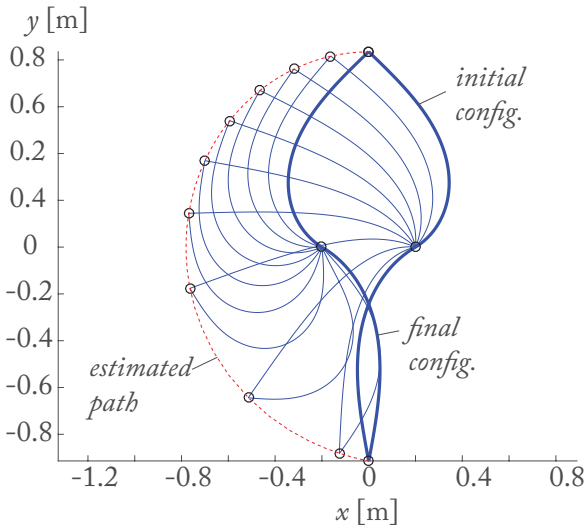


Fig. 4. Path estimated for the $\underline{R}FRF\underline{R}$ robot modelled with 5 modes per leg (to scale).

TABLE I
MEAN, MINIMAL AND MAXIMAL ESTIMATION ERRORS W.R.T. MOD. #2 FOR THE END-EFFECTOR POSITION AND FOR THE MOTOR ANGLE POSITIONS AS A FUNCTION OF THE NUMBER OF MODES PER LEG IN MOD. #1, COMPUTED ON 51 CONFIGURATIONS FOR THE $\underline{R}FRF\underline{R}$ ROBOT.

	Position error [mm]			Motor angle error [mdeg]		
Nb. of modes	Min	Max	Mean	Min	Max	Mean
3	0.11	9.22	1.50	13.75	18.50	16.13
4	4.5e-3	2.68	0.68	3.10	3.18	3.14
5	6.5e-5	0.13	0.029	1.40e-2	0.15	8.12e-2

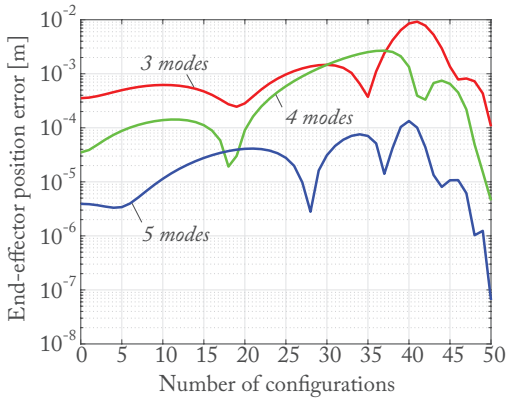
such as [67]–[69], and comparing the results of our modelling approach with it is worthy of investigation. Results are shown in Fig. 5(b), and a summary of the results is presented in Table II. For having an error similar the error obtained with our approach with 4 modes per leg only, 1000 elements per leg (and thus, 2000 elastic variables) are necessary, which leads to system of equations of large dimension which are time consuming to solve.

We finally compared the validity of the inverse geometrico-static model, i.e. the error of prediction of the motor angle values with respect to Mod. #2, along a path followed by the end-effector defined by:

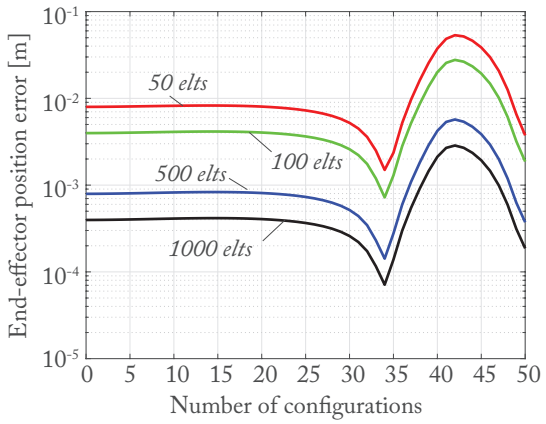
$$\mathbf{q}_{\mathrm{p}}(t) = \frac{t}{n_c}(\mathbf{q}_{\mathrm{pf}} - \mathbf{q}_{\mathrm{p0}}) + \mathbf{q}_{\mathrm{p0}} \tag{54}$$

where $\mathbf{q}_{\mathrm{p0}}$ ($\mathbf{q}_{\mathrm{pf}}$, resp.) is the end-effector initial (final, resp.) configuration. Here, we took $\mathbf{q}_{\mathrm{p0}}$ and $\mathbf{q}_{\mathrm{pf}}$ as the initial and final configurations of corresponding to $\mathbf{q}_{\mathrm{a0}}$ and $\mathbf{q}_{\mathrm{af}}$, respectively. Results are provided in Tab. I. With five modes per legs, the prediction error of our model is lower than 1.5e-4 degrees, which is more than acceptable.

2) Computational time: We also want to analyze the computational performance of our model. For this, we fed the forward geometrico-model (our model (Mod #1) with 5 modes per leg, the continuous Cosserat model [21] (Mod #2) and the finite difference approach (Mod #3) with 1000 elements per leg) with the equation (53) for varying number of configu-



(a) Error in the estimation of the end-effector position for our model as a function of the number of modes per leg



(b) Error in the estimation of the end-effector position for the finite difference approach as a function of the number of element per leg

Fig. 5. Error in the estimation of the end-effector position for our approach and for the finite difference approach [44] w.r.t. the continuous Cosserat model [21] for 51 configurations of the $\underline{R}FRF\underline{R}$ robot spread along the path shown in Fig. 4.

TABLE II
MEAN, MINIMAL AND MAXIMAL ESTIMATION ERRORS W.R.T. MOD. #2 FOR THE END-EFFECTOR POSITION AS A FUNCTION OF THE NUMBER OF FINITE ELEMENTS IN MOD. #3, COMPUTED ON 51 CONFIGURATIONS FOR THE $\underline{R}FRF\underline{R}$ ROBOT.

	Position error [mm]		
Nb. of elts.	Min	Max	Mean
50	1.50	53.44	12.94
100	0.72	27.76	6.59
500	0.14	5.71	1.34
1000	0.07	2.86	0.67

rations n_c and we recorded their durations for providing the results in terms of end-effector configuration. For making a fair comparison, for all models tested in this section:

- The initial configuration is already known and is used as the first initial guess of the solver.
- For the other configurations, the initial guess is a predictor based on the knowledge of (i) the configuration computed at the step before, (ii) the model Jacobian matrix whose nullspace can be used in order to predict a variation of the model variables in the next configuration.
- All computations are run 10 times and the computational time provided is the mean time.
- The analytical Jacobian matrices are implemented in both approaches in order to speed up the computation.
- The setting parameters for the Matlab function *ode45* are the "by-default" ones are are identical for all approaches. Moreover, for the *fsolve* function, the stopping parameters are also set as identical for all models.

TABLE III
TOTAL COMPUTATIONAL TIMES FOR THE $n_c + 1$ CONFIGURATIONS WITH OUR MODEL (MOD #1, 5 MODES PER LEG), THE MODEL [21] (MOD #2) AND FINITE DIFFERENCES (MOD #3, 1000 ELEMENTS PER LEG) FOR THE $\underline{R}FRF\underline{R}$ ROBOT.

n_c	Mod #1 [s]	Mod #2 [s]	Mod #3
50	2.8	1.9	325
30	1.6	1.1	202
15	0.8	0.7	105
10	0.6	0.4	87
5	0.3	0.3	57

Results are shown in Tab. III. Globally, our model with 5 modes per leg (i.e. with a mean accuracy of 30 microns) is as computational efficient as the continuous Cosserat model. For a lower accuracy of prediction, the computational time of the finite differences approach is already more than 100 time bigger.

3) Checking stability: In this section, we compare the prediction of stability based on the analysis of the spectrum of the matrix $\mathbf{H}_r$ defined in Section V-D with the criterion defined in the work [48]. In this latter context, the stability of the configuration is assessed by looking at the determinant of a matrix denoted as $\mathbf{b}_\lambda(u)$ which must be computed all along an integration interval $u \in [0, 1[$. If this determinant is null for $u = u_{cp} \in [0, 1[$, a so-called conjugate point appears, which is a condition of unstability. We do not provide any further details on the computation of the matrix $\mathbf{b}_\lambda(u)$ and refer the reader to [48] for any details of implementation.

In Figure 6, we show the robot end-effector configuration space computed with the flooding algorithm proposed in [33] for the leg buckling modes shown in Fig. 3. In this picture, we highlight in red the areas where the inverse condition number of the matrix $\mathbf{H}_r$ in (51) is lower than 10^{-4}, i.e. near which stability issues may appear. Note that this value of 10^{-4} has been fixed in order to have a better display of the results in Fig. 6. For checking the prediction of stability based on our model, we define a path between points $X_1 = (0,\ 0.8)$ m and $X_5 = (0,\ -0.2)$ m along which the stability criterion defined in the work [48] will be computed. Along this path, 50 points are defined. In Fig. 7, the minimal value of $\det(\mathbf{b}_\lambda)$ for each point is plotted, in parallel as the smallest eigenvalue σ_1 of the matrix $\mathbf{H}_r$. From Figs. 7 and 8, we observe the following things:

- From point X_1 ($\sigma_1 > 0$, $\sigma_2 > 0$) to point X_2 ($\sigma_1 = 0$, $\sigma_2 > 0$, excluded) on the singularity of $\mathbf{H}_r$ (see Fig. 6), there is no conguate point
- At point X_2 ($\sigma_1 = 0$, $\sigma_2 > 0$), a conjugate point appears on the integration interval at $u = 0$, meaning that we

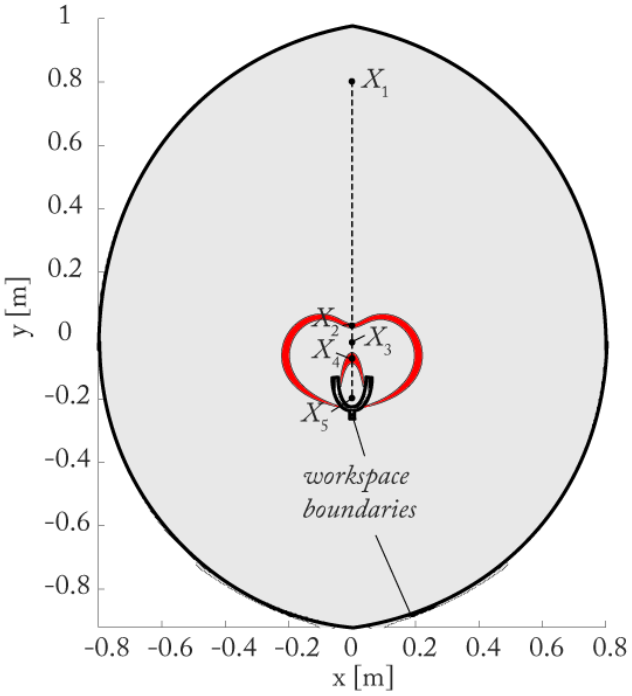


Fig. 6. Path tested in the workspace.

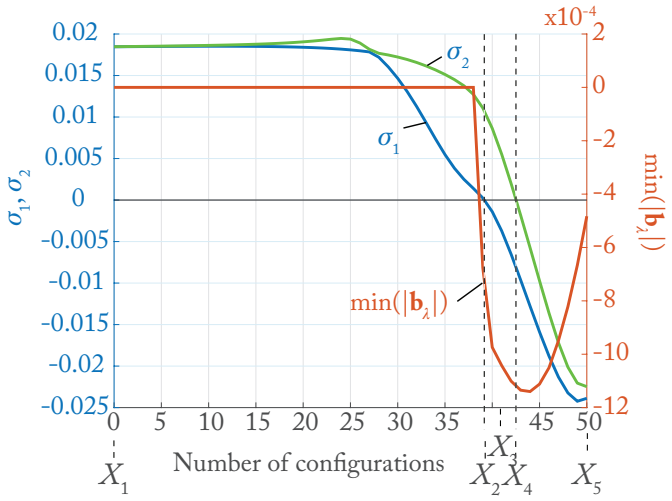


Fig. 7. Comparison stability criterion from [48] and the two first smallest eigenvalues σ_1 and σ_2 of the reduced Hessian matrix $\mathbf{H}_r$.

reached a limit of stability.

- From point X_2 ($\sigma_1 = 0$, $\sigma_2 > 0$) to point X_4 ($\sigma_1 < 0$, $\sigma_2 = 0$, excluded), a single conjugate point exists on the interval $u \in [0\ 1[$, meaning that the robot is instable along this path.
- At point X_4 ($\sigma_1 < 0$, $\sigma_2 = 0$), a second conjugate point appear, meaning that we cross a second zone of singularity of $\mathbf{H}_r$.
- From point X_4 ($\sigma_1 < 0$, $\sigma_2 = 0$) to point X_5 ($\sigma_1 < 0$, $\sigma_2 < 0$), two conjugate points exist on the interval $u \in [0\ 1[$, meaning that the robot has two degrees of unstability.

To summarize, every time an eigenvalue of the Hessian matrix crosses zero, a conjugate point appears.

B. The spatial $6-\underline{R}FS$ robot

Here, we study a continuum spatial parallel $6-\underline{R}FS$ robot made of six rods (Fig. 9). Each rod is connected at the ground via an actuated revolute joint (points A_i). Its extremity is linked to a rigid moving platform via a passive spherical joint (S joint at points B_i). Because the robot has six motors, it is possible to control the position and orientation of the frame $\mathcal{F}_P : (P, \boldsymbol{x}_P, \boldsymbol{y}_P, \boldsymbol{z}_P)$.

In the base frame $\mathcal{F}_0 : (O, \boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z})$, positions of points A_i are given by: $\overrightarrow{OA_i} = r_b \left[\cos\gamma_i \ \sin\gamma_i \ 0\right]^T$ $(i = 1, 2, 3)$ with $r_b = 0.25$ m and $\gamma_1 = 0$, $\gamma_2 = 2\pi/3$, and $\gamma_3 = -2\pi/3$. In the platform frame $\mathcal{F}_P$, positions of points B_i are given by: $\overrightarrow{PB_i} = r_p \left[\cos\alpha_i \ \sin\alpha_i \ 0\right]^T$ $(i = 1, 2, 3)$ with $r_p = 0.1$ m and $\alpha_1 = \pi/3$, $\alpha_2 = \pi$, and $\alpha_3 = -\pi/3$. Parameters of the rods are: rods at rest are straight, their length L is equal to $L = 1$ m, that have circular cross-sections of radius 1 mm, Young's and shear moduli $E = 210$ GPa and $G = 82$ GPa, respectively, and density $\rho = 7800$ kg/m^3. No external wrenches are applied.

We used Matlab for encoding the robot both with our model taking the same number of modes m_j for the bending per leg (in what follows, $m_j = 3$, 4 or 5) and with the continuous Cosserat model [21] (assuming spatial Kirchhoff hypotheses with no extensibility). It should be mentioned that, due to the presence of passive spherical joints, no torsion is transmitted inside the legs. Therefore, no torsion deformation modes have been included into the model.

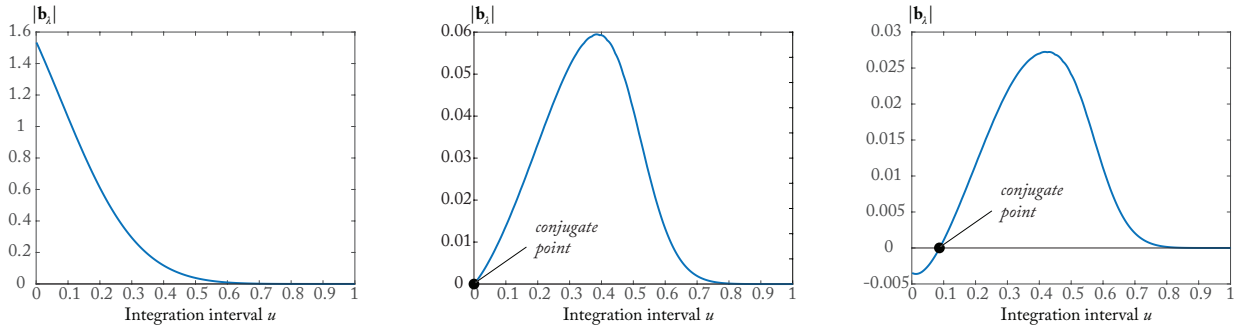
We will then again compare our model to the approach [21] in terms of accuracy of prediction for the geometrico-static model, computational time and stability prediction.

It should be mentioned that, as previously, for all computations shown thereafter (for both our model and the model [21]), the solver used is the Levenberg-Marquardt algorithm encoded in the Matlab *fsolve* function.

1) Model prediction accuracy: For checking the model accuracy, we used the same approach as in Section VI-A. We first estimate the position of the end-effector both with our model and the model in [21] along a path followed by the motors defined by the Eq. (53). Here, we took $\mathbf{q}_{a0} = [5.86\ 5.56\ 5.86\ 5.46\ 5.86\ 5.26]^T$ rad, $\mathbf{q}_{af} = [7.95\ 7.65\ 4.29\ 3.89\ 3.77\ 3.17]^T$ rad. An example of computed configurations along this path for $n_c = 10$ is shown in Fig. 10.

The error of prediction between our model (denoted as Mod. #1) and the model [21] (denoted as Mod. #2) for the end-effector position for $n_c + 1 = 51$ configurations computed by feeding the forward geometrico-static models with the equation (53) is shown in Fig. 11(a), and a summary of the results (for both position and orientation errors) is presented in Table IV. It should be mentioned that the orientation error is calculated as the norm of the vector $\left(\mathbf{R}_{\text{Mod1}}^T\mathbf{R}_{\text{Mod2}} - \mathbf{R}_{\text{Mod2}}^T\mathbf{R}_{\text{Mod1}}\right)^\vee$, where $\mathbf{R}_{\text{Mod1}}$ is the platform orientation matrix computed with our approach, and $\mathbf{R}_{\text{Mod2}}$ the platform orientation matrix computed with [21].

Note that we did not compare our approach with finite



(a) $|\mathbf{b}_\lambda|$ at X_1 ($\sigma_1 > 0$, $\sigma_2 > 0$) (b) $|\mathbf{b}_\lambda|$ at X_2 ($\sigma_1 = 0$, $\sigma_2 > 0$) (c) $|\mathbf{b}_\lambda|$ at X_3 ($\sigma_1 < 0$, $\sigma_2 > 0$)

(d) $|\mathbf{b}_\lambda|$ at X_4 ($\sigma_1 < 0$, $\sigma_2 = 0$) (e) $|\mathbf{b}_\lambda|$ at X_5 ($\sigma_1 < 0$, $\sigma_2 < 0$)

Fig. 8. Stability criterion from [48] at points X_1, X_2, X_3, X_4 and X_5.

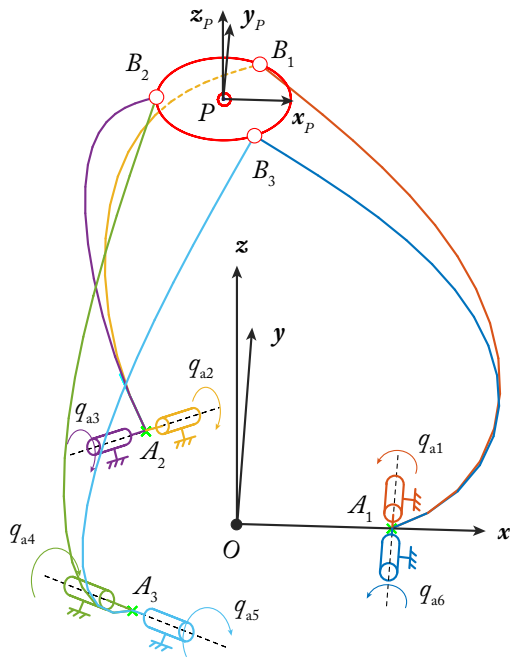


Fig. 9. Schematics of the $6-\underline{R}FS$ robot (to scale).

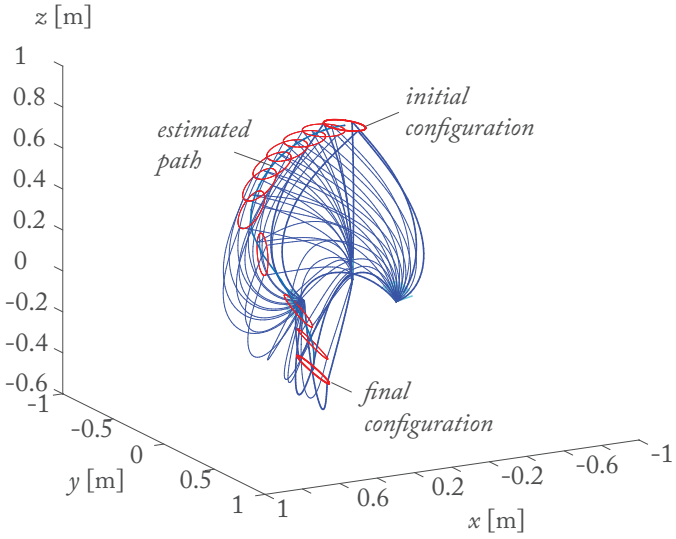
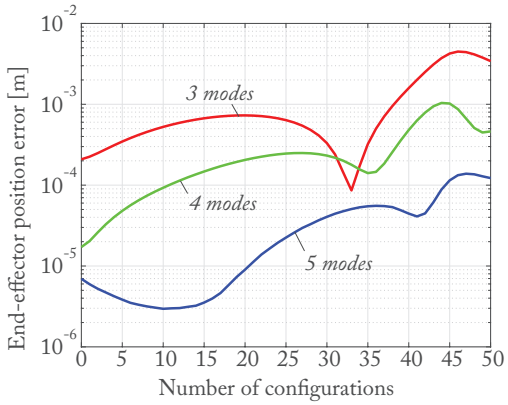


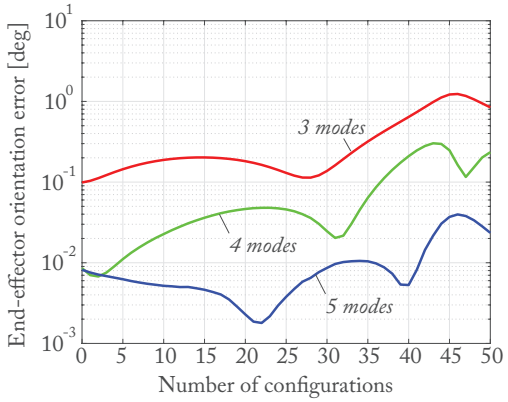
Fig. 10. Path estimated for the $6-\underline{R}FS$ robot modelled with 5 modes per leg (to scale).

differences in the spatial case. Indeed, as shown in [33], for finite differences, 500 elements per leg in the spatial case (leading to 2000 variables per leg for modeling the elasticity) lead to an error of end-effector position estimation of 1 mm. However, due to the large size of the equation system (more than 12,000 equations), the computational time for a single configuration is bigger than 10 minutes. Therefore, in our opinion, further comparisons with our model and finite element approach were not relevant.

Again, as expected, the prediction error decreases with the number of modes per leg. With five modes per bending per leg, i.e. a total of 10 modes per leg (60 components in the vector $\mathbf{q}_e$ in totality), the mean position estimation error is lower than 40 microns, while the mean angular estimation error is lower than 0.01 deg., which is usually much enough for a robot with



(a) Error in the estimation of the end-effector position as a function of number of modes per bending per leg



(b) Error in the estimation of the end-effector orientation as a function of number of modes per bending per leg

Fig. 11. Error in the estimation of the end-effector position orientation for our approach w.r.t. the continuous Cosserat model [21] for 51 configurations of the $6-\underline{R}FS$ robot spread along the path shown in Fig. 10.

TABLE IV
MEAN, MINIMAL AND MAXIMAL ESTIMATION ERRORS FOR THE END-EFFECTOR POSITION AS A FUNCTION OF THE NUMBER OF MODES COMPUTED ON 51 CONFIGURATIONS OF THE $6-\underline{R}FS$ ROBOT.

	Position err. [mm]			Angular err. [deg]		
Modes	Min	Max	Mean	Min	Max	Mean
3	0.09	4.49	1.14	0.10	1.24	0.37
4	0.02	1.04	0.28	6.8e-3	0.30	0.08
5	3.0e-4	0.14	3.6e-2	1.8e-3	0.04	0.01

legs of 1 m.

Finally, again, we checked the validity of the inverse geometrico-static model, i.e. the accuracy of prediction for the motor position with respect to Mod. #2, knowing the end-effector pose. Here, we took the 51 end-effector configurations computed with Mod. #2 as inputs of the inverse geometrico-static model of Mod. #1 with five modes, and we computed the motor angles. Results are shown in Tab. V. Results show that the mean value of the errors of prediction is below 0.02 degrees, which is more than acceptable. A peak value of 0.17 degrees is obtained for the 6th motor of the robot, which is due to the presence of a singularity near the computed pose. Far from singularities, the accuracy of prediction is considerably better.

TABLE V
MEAN, MINIMAL AND MAXIMAL ESTIMATION ERRORS FOR THE MOTOR ANGLE POSITION AND ORIENTATION AS A FUNCTION OF THE NUMBER OF MODES COMPUTED ON 51 CONFIGURATIONS OF THE $6-\underline{R}FS$ ROBOT, WITH 5 MODES.

	Angular error [mdeg]					
	Mot#1	Mot#2	Mot#3	Mot#4	Mot#5	Mot#6
Min	1.5e-2	0.33	8.9e-4	3.7e-4	9.2e-4	4.1e-3
Max	52.22	36.59	31.59	13.80	3.68	176.84
Mean	10.58	4.17	2.47	1.62	0.77	19.06

TABLE VI
TOTAL COMPUTATIONAL TIMES FOR THE n_c+1 CONFIGURATIONS WITH OUR MODEL (MOD #1, 5 MODES), THE MODEL [21] (MOD #2) FOR THE $6-\underline{R}FS$ ROBOT.

n_c	Mod #1 [s]	Mod #2 [s]
50	130	191
30	80	124
15	46	62
10	27	41
5	13	26

2) Computational time: We want to analyze the computational performance of our model, adopting the same strategy as in as in Section VI-A. Again, we fed the forward geometrico-model (our model with 5 modes per bending per length and the model [21]) with the equation (53) for varying number of configurations n_c and we recorded their durations for providing the results in terms of end-effector configuration. For making a fair comparison between both models tested in this section, we used the same assumptions as in Section VI-A2.

Results are shown in Tab. VI and we may conclude that our model is computational efficient.

3) Checking stability: As in Section VI-A3, we compare the prediction of stability based on the analysis of the spectrum of the matrix $\mathbf{H}_r$ defined in Section V-D with the criterion defined in the work [48].

In Figure 12, we show a slice in xy (for $z = 0.635$ m, $\phi = \pi/8$ rad, $\theta = \pi/3$ rad, and $\psi = 0$ rad, these three angles being defined according to a ZYZ sequence of Euler-angles) of the robot end-effector configuration space computed with the flooding algorithm proposed in [33] for the leg buckling modes shown in Fig. 9. In this picture, we highlight in red the areas where the inverse condition number of the matrix $\mathbf{H}_r$ in (51) is lower than $5 \cdot 10^{-4}$, i.e. close to the limit beyond which, stability issues may occur. For checking the prediction of stability based on our model, we define a path between points $X_1 = (-0.4,\ 0)$ m and $X_3 = (-0.4,\ 0.2)$ m (for $z = 0.635$ m, $\phi = \pi/8$ rad, $\theta = \pi/3$ rad, and $\psi = 0$ rad) along which the stability criterion defined in the work [48] (presence of conjugate points) is computed. Along this path, 50 points are defined. In Fig. 13, the minimal value of $\det(\mathbf{b}_\lambda)$ for each point is plotted, in parallel as the smallest eigenvalue σ_1 of the matrix $\mathbf{H}_r$. From Figs. 13 and 14, we observe the following facts:

- From point X_1 ($\sigma_1 > 0$) to point X_2 ($\sigma_1 = 0$, excluded) on the singularity of $\mathbf{H}_r$ (see Fig. 12), there is no conjugate point.

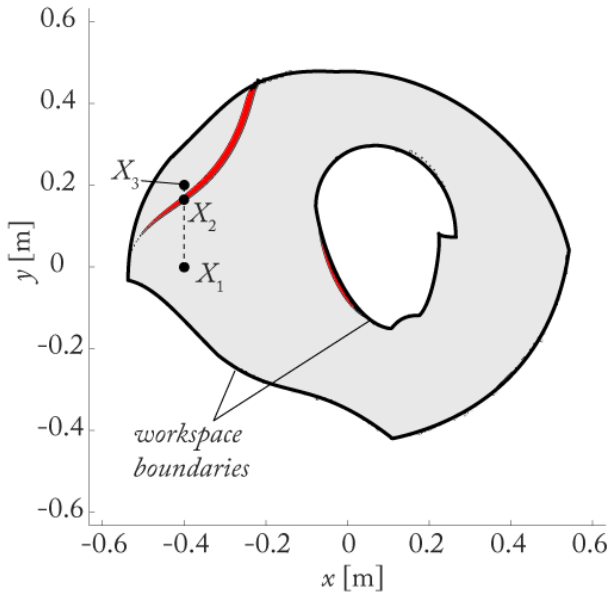


Fig. 12. Path tested in the workspace, at $z = 0.635$ m, $\phi = \pi/8$ rad, $\theta = \pi/3$ rad, and $\psi = 0$ rad, these three angles being defined in the ZYZ Euler-angle convention.

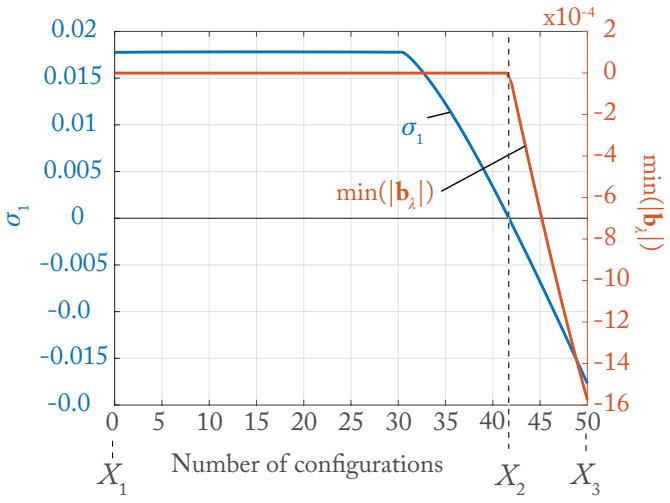


Fig. 13. Comparison stability criterion from [48] and the smallest eigenvalue σ_1 of the reduced Hessian matrix $\mathbf{H}_r$.

- At point X_2 ($\sigma_1 = 0$), a conjugate point appears on the integration interval at $u = 0$, meaning that we reached a limit of stability.
- From point X_2 ($\sigma_1 = 0$) to point X_3 ($\sigma_1 < 0$), a conjugate point exists on the interval $u \in [0\ 1[$, meaning that the robot is unstable along this path.

All these results showed the interest of our modelling approach.

VII. CONCLUSIONS

Several works have focused on the computation of geometrico- and kinemato-static models of CPRs. Those works can be sorted into two main categories: (i) models based on the continuous Cosserat equations and (ii) discretized models. The first types of models are very accurate but assessing elastic stability with them is a complicated task. The second types of models allow easily checking the stability but they require a large number of elastic variables to be accurate.

In this paper, we extended an approach based on assumed strain modes that was developed for the dynamics of serial continuum robots to the statics of CPRs. We showed that our model give very similar results than those obtained with the full continuous Cosserat model, and this, with a very limited number of elastic variables, contrary to other standard discretization methods: Typically, 10 elastic variables per leg lead to a prediction accuracy of 50 microns for 1-meter-length rods. The method was not only accurate but also computationally efficient: For a better prediction accuracy than discretization based on finite-differences, it was more than 100 times faster. Furthermore, the computational time was similar to that for the continuous Cosserat model. Finally, it allows the elastic stability to be assessed, by only checking the reduced Hessian of the potential energy as for any discrete Lagrangian model. This reduced Hessian can be computed by using matrices already obtained for the calculation of the robot kinemato-static model, thus making the analysis of this robot property simpler than for the continuous Cosserat model. All the results have been validated on two case studies: a planar $\underline{R}FRF\underline{R}$ robot and a spatial $6-\underline{R}FS$ robot.

Our future works will deal with the extension of the method to the dynamics of CPRs.

APPENDIX

A. Recalls of Lie group notations

Some notational conventions of Lie group theory, that can be found in [57], are recalled here for reasons of convenience. A hat "$\wedge$" covering a vector $\boldsymbol{\Upsilon}$ defines a matrix $\hat{\boldsymbol{\Upsilon}}$ whose expression depends on the dimension of $\boldsymbol{\Upsilon}$. If $\boldsymbol{\Upsilon} \in \mathbb{R}^3$, then $\hat{\boldsymbol{\Upsilon}} = \boldsymbol{\Upsilon}^\wedge$ denotes the (3×3) skew symmetric matrix defined such that: $\hat{\boldsymbol{\Upsilon}}\mathbf{x} = \boldsymbol{\Upsilon} \times \mathbf{x}$ for any $\mathbf{x} \in \mathbb{R}^3$. If $\boldsymbol{\Upsilon} = [\mathbf{a}^T\ \mathbf{b}^T]^T \in \mathbb{R}^6$, with $\mathbf{a},\ \mathbf{b} \in \mathbb{R}^3$, then $\hat{\boldsymbol{\Upsilon}}$ is the (4×4) matrix defined by:

$$\hat{\boldsymbol{\Upsilon}} = \begin{bmatrix} \hat{\mathbf{a}} & \mathbf{b} \\ \mathbf{0}_{1\times 3} & 0 \end{bmatrix} \tag{55}$$

Reciprocally, the superscript "$\vee$" is such that $\hat{\boldsymbol{\Upsilon}}^\vee = \boldsymbol{\Upsilon}$ for any $\boldsymbol{\Upsilon} \in \mathbb{R}^3$ or $\mathbb{R}^6$. The two operators $\mathbf{Ad}$ and $\mathbf{ad}$ are respectively defined for $\boldsymbol{g} = (\mathbf{R},\ \mathbf{p}) \in SE(3)$ and $\boldsymbol{\Upsilon} = [\mathbf{a}^T\ \mathbf{b}^T]^T$ by the two (6×6) matrices

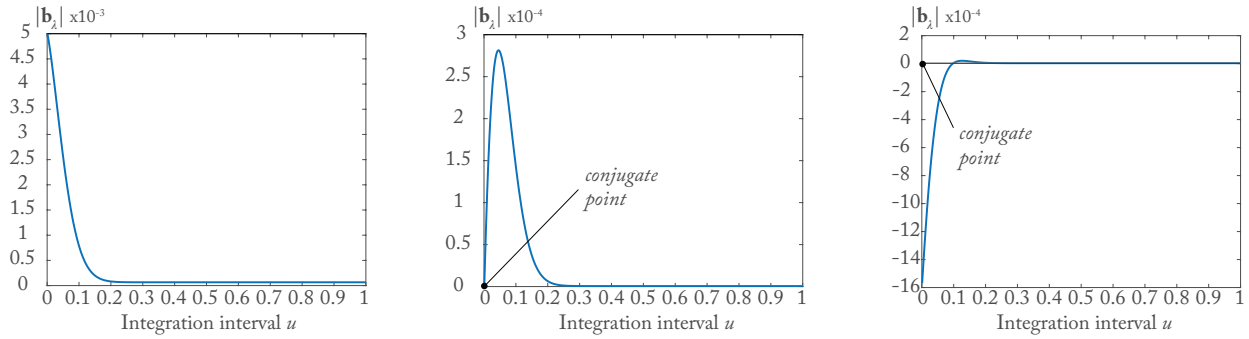
$$\mathbf{Ad}_{\boldsymbol{g}} = \begin{bmatrix} \mathbf{R} & \mathbf{0} \\ \hat{\mathbf{p}}\mathbf{R} & \mathbf{R} \end{bmatrix}, \quad \mathbf{ad}_{\boldsymbol{\Upsilon}} = \begin{bmatrix} \hat{\mathbf{a}} & \mathbf{0} \\ \hat{\mathbf{b}} & \hat{\mathbf{a}} \end{bmatrix} \tag{56}$$

Note that $\mathbf{Ad}_{\boldsymbol{g}}$ is the twist transformation matrix allowing to pass from one frame to another one, their relative pose being parameterized by the transformation $\boldsymbol{g} = (\mathbf{R},\ \mathbf{p})$. Note also that, if $\boldsymbol{\Upsilon} = (\boldsymbol{g}^{-1}\boldsymbol{g}')^\vee$, then we have:

$$\mathbf{Ad}'_{\boldsymbol{g}} = \mathbf{Ad}_{\boldsymbol{g}}\mathbf{ad}_{\boldsymbol{\Upsilon}} \Leftrightarrow \mathbf{ad}_{\boldsymbol{\Upsilon}} = \mathbf{Ad}_{\boldsymbol{g}}^{-1}\mathbf{Ad}'_{\boldsymbol{g}} \tag{57}$$

Finally, let us define two vectors $\boldsymbol{\Upsilon}_1$ and $\boldsymbol{\Upsilon}_2$ in $\mathbb{R}^6$. Then,

$$(\hat{\boldsymbol{\Upsilon}}_1\hat{\boldsymbol{\Upsilon}}_2 - \hat{\boldsymbol{\Upsilon}}_2\hat{\boldsymbol{\Upsilon}}_1)^\vee = \mathbf{ad}_{\boldsymbol{\Upsilon}_1}\boldsymbol{\Upsilon}_2 = -\mathbf{ad}_{\boldsymbol{\Upsilon}_2}\boldsymbol{\Upsilon}_1 \tag{58}$$



(a) $|\mathbf{b}_\lambda|$ at X_1 ($\sigma_1 > 0$) (b) $|\mathbf{b}_\lambda|$ at X_2 ($\sigma_1 = 0$) (c) $|\mathbf{b}_\lambda|$ at X_3 ($\sigma_1 < 0$)

Fig. 14. Stability criterion from [48] at points X_1, X_2 and X_3.

This result can be easily verified by assigning symbolic values to the components of $\boldsymbol{\Upsilon}_1$ and $\boldsymbol{\Upsilon}_2$ and developing all sides of the equation (58).

REFERENCES

[1] S. Hirose and M. Mori, "Biologically inspired snake-like robots," in *Proceedings of the 2004 IEEE International Conference on Robotics and Biomimetics (ROBIO2004)*, Shenyang, China, 2004.

[2] D. Camarillo, C. Milne, C. Carlson, M. Zinn, and J. Salisbury, "Mechanics modeling of tendon-driven continuum manipulators," *IEEE Transactions on Robotics*, vol. 24, no. 6, pp. 1262–1273, 2008.

[3] I. Gravagne, C. Rahn, and I. Walker, "Large deflection dynamics and control for planar continuum robots," *IEEE/ASME Transactions on Mechatronics*, vol. 8, no. 2, pp. 299–307, 2003.

[4] J. Edelmann, A. Petruska, and B. Nelson, "Estimation-based control of a magnetic endoscope without device localization," *Journal of Medical Robotics Research*, vol. 3, no. 1, 2017.

[5] ——, "Magnetic control of continuum devices," *The International Journal of Robotics Research*, vol. 36, no. 1, pp. 68–85, 2017.

[6] M. Ikeuchi and K. Ikuta, "Development of pressure-driven micro active catheter using membrane micro emboss following excimer laser ablation (MeME-x) process," in *2009 IEEE International Conference on Robotics and Automation*. IEEE, may 2009.

[7] B. Jones and I. Walker, "Kinematics for multisection continuum robots," *IEEE Transactions on Robotics*, vol. 22, no. 1, pp. 43–55, feb 2006.

[8] ——, "Practical kinematics for real-time implementation of continuum robots," *IEEE Transactions on Robotics*, vol. 22, no. 6, pp. 1087–1099, dec 2006.

[9] M. Langelaar and F. Keulen, "Modeling of a shape memory alloy active catheter," *45th AIAA Structures, Structural Dynamics and Materials Conference*, pp. 19–22, 2004.

[10] S. Lee, S. Lee, H. An, S. Cha, J. Chang, B. Kim, and J. Pak, "Biomedical applications of electroactive polymers and shape memory alloys," *Proceedings of SPIE-The International Society for Optical Engineering*, vol. 4695, pp. 17–31, 2002.

[11] K. Lee, N. Munce, T. Shoa, L. Charron, G. Wright, J. Madden, and V. Yang, "Fabrication and characterization of laser-micromachined polypyrrole-based artificial muscle actuated catheters," *Sensors and Actuators A: Physical*, vol. 153, no. 2, pp. 230 – 236, 2009.

[12] K. Kim and S. Tadokoro, Eds., *Electroactive Polymers for Robotic Applications*. Springer London, 2007.

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

[14] M. Chikhaoui, K. Rabenorosoa, and N. Andreff, "Kinematics and performance analysis of a novel concentric tube robotic structure with embedded soft micro-actuation," *Mechanism and Machine Theory*, vol. 104, pp. 234 – 254, 2016.

[15] J. Webster, J. Romano, and N. Cowan, "Mechanics of precurved-tube continuum robots," *IEEE Transactions on Robotics*, vol. 25, no. 1, pp. 67–78, 2009.

[16] N. Simaan, "Snake-like units using flexible backbones and actuation redundancy forenhanced miniaturization," in *Proceedings of the 2005 IEEE International Conference on Robotics and Automation (ICRA 2005)*, Barcelona, Spain, 2005, pp. 3012–3017.

[17] K. Xu and N. Simaan, "Analytic formulation for kinematics, statics and shape restoration of multibackbone continuum robots via elliptic integrals," *ASME Journal of Mechanisms and Robotics*, vol. 2, no. 1, pp. 011 006–1–011 006–13, 2010.

[18] L. Wang, G. Del Giudice, and N. Simaan, "Simplied kinematics of continuum robot equilibrium modulation via moment coupling effects and model calibration," *ASME Journal of Mechanisms and Robotics*, vol. 11, no. 5, 2019.

[19] J. Burgner-Kahrs, D. Rucker, and H. Choset, "Continuum robots for medical applications: A survey," *IEEE Transactions on Robotics*, vol. 31, no. 6, pp. 1261–1280, 2015.

[20] C. Bryson and D. Rucker, "Toward parallel continuum manipulators," in *Proceedings of the 2014 IEEE International Conference on Robotics and Automation (ICRA 2014)*, 2014.

[21] C. Black, J. Till, and D. Rucker, "Parallel continuum robots: Modeling, analysis, and actuation-based force sensing," *IEEE Transactions on Robotics*, vol. 34, no. 1, pp. 29–47, 2017.

[22] J. Merlet, *Parallel Robots*, 2nd ed. Springer, 2006.

[23] E. M. Young and K. J. Kuchenbecker, "Implementation of a 6-DOF parallel continuum manipulator for delivering fingertip tactile cues," *IEEE Transactions on Haptics*, vol. 12, no. 3, pp. 295–306, jul 2019.

[24] C. Pacchierotti, E. M. Young, and K. J. Kuchenbecker, "Task-driven PCA-based design optimization of wearable cutaneous devices," *IEEE Robotics and Automation Letters*, vol. 3, no. 3, pp. 2214–2221, jul 2018.

[25] O. Altuzarra, M. Diez, J. Corral, G. Teoli, and M. Ceccarelli, "Kinematic analysis of a continuum parallel robot," in *New Trends in Mechanism and Machine Science*. Springer International Publishing, aug 2016, pp. 173–180.

[26] O. Altuzarra, D. Caballero, F. Campa, and C. Pinto, "Forward and inverse kinematics in 2-dof planar parallel continuum manipulators," in *Proceedings of the European Conference on Mechanism Science (EuCoMeS 2018)*, Aachen, Germany, 2018.

[27] O. Altuzarra, D. Caballero, Q. Zhang, and F. Campa, "Kinematic characteristics of parallel continuum mechanisms," in *Proceedings of Advances in Robot Kinematics (ARK 2018)*, Bologna, Italy, 2018.

[28] O. Altuzarra and J. Merlet., "Certified kinematics solution of 2-dof planar parallel continuum mechanisms," in *Proceedings of the IFToMM World Congress on Mechanism and Machine Science*, Cracow, Poland, 2019, pp. 197–208.

[29] O. Altuzarra, D. Caballero, F. Campa, and C. Pinto, "Position analysis in planar parallel continuum mechanisms," *Mechanism and Machine Theory*, vol. 132, pp. 13–29, 2019.

[30] F. Campa, M. Diez, D. Diaz-Caneja, and O. Altuzarra, "A 2 dof continuum parallel robot for pick & place collaborative tasks," in *Advances in Mechanism and Machine Science*. Springer International Publishing, 2019, pp. 1979–1988.

[31] B. Mauze, R. Dahmouche, G. Laurent, N. Andreff, P. Rougeot, P. Sandoz, and C. Clevy, "Nanometer precision with a planar parallel continuum robot," *IEEE Robotics and Automation Letters*, vol. 5, no. 3, pp. 3806–3813, jul 2020.

[32] Z. Yang, X. Zhu, and K. Xu, "Continuum delta robot: a novel translational parallel robot with continuum joints," in *2018 IEEE/ASME International Conference on Advanced Intelligent Mechatronics (AIM)*. IEEE, jul 2018.

[33] S. Briot and A. Goldsztejn, "Singularity conditions for continuum parallel robots," *IEEE Transactions on Robotics*, pp. 1–19, 2021, in Early Access.

[34] G. Chen, Y. Kang, Z. Liang, Z. Zhang, and H. Wang, "Kinetostatics modeling and analysis of parallel continuum manipulators," *Mechanism and Machine Theory*, vol. 163, p. 104380, sep 2021.

[35] S. Lilge, K. Nuelle, G. Boettcher, S. Spindeldreier, and J. Burgner-Kahrs, "Tendon actuated continuous structures in planar parallel robots: A kinematic analysis," *Journal of Mechanisms and Robotics*, vol. 13, no. 1, dec 2020.

[36] K. Nuelle, T. Sterneck, S. Lilge, D. Xiong, J. Burgner-Kahrs, and T. Ortmaier, "Modeling, calibration, and evaluation of a tendon-actuated planar parallel continuum robot," *IEEE Robotics and Automation Letters*, vol. 5, no. 4, pp. 5811–5818, oct 2020.

[37] G. Boettcher, S. Lilge, and J. Burgner-Kahrs, "Design of a reconfigurable parallel continuum robot with tendon-actuated kinematic chains," *IEEE Robotics and Automation Letters*, vol. 6, no. 2, pp. 1272–1279, apr 2021.

[38] P. Anderson, A. Mahoney, and R. Webster, "Continuum reconfigurable parallel robots for surgery: Shape sensing and state estimation with uncertainty," *IEEE Robotics and Automation Letters*, vol. 2, no. 3, pp. 1617–1624, jul 2017.

[39] A. L. Orekhov, V. A. Aloi, and D. C. Rucker, "Modeling parallel continuum robots with general intermediate constraints," in *2017 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, may 2017.

[40] G. Wu and G. Shi, "Experimental statics calibration of a multi-constraint parallel continuum robot," *Mechanism and Machine Theory*, vol. 136, pp. 72–85, jun 2019.

[41] K. Stoy, K. Walker, S. Nielsen, P. Ayres, M. K. Heinrich, D. Leon, and A. Cheheltan, "A large-scale, light-weight, and soft braided robot manipulator with rapid expansion capabilities," in *Proceedings of the 4th IEEE International Conference on Soft Robotics (RoboSoft 2021)*, Yale University, USA, 2021, pp. 495–500.

[42] G. Wu and G. Shi, "Design, modeling, and workspace analysis of an extensible rod-driven parallel continuum robot," *Mechanism and Machine Theory*, vol. 172, pp. 104 798–1–104 798–16, 2022.

[43] R. J. Webster and B. A. Jones, "Design and kinematic modeling of constant curvature continuum robots: A review," *The International Journal of Robotics Research*, vol. 29, no. 13, pp. 1661–1683, 2010.

[44] F. Zaccaria, S. Briot, M. Chikhaoui, E. Idà, and M. Carricato, "An analytical formulation for the geometrico-static problem of continuum planar parallel robots," in *Proceedings of the 23rd CISM IFToMM Symposium on Robot Design, Dynamics and Control (RoManSy 2020)*, 2020.

[45] A. Orekhov, C. Black, J. Till, S. Chung, and D. Rucker, "Analysis and validation of a teleoperated surgical parallel continuum manipulator," *IEEE Robotics and Automation Letters*, vol. 1, no. 2, pp. 828–835, jul 2016.

[46] I. Singh, M. Singh, P. M. Pathak, and R. Merzouki, "Optimal work space of parallel continuum manipulator consisting of compact bionic handling arms," in *2017 IEEE International Conference on Robotics and Biomimetics (ROBIO)*. IEEE, dec 2017.

[47] F. Zaccaria, E. Idà, S. Briot, and M. Carricato, "Workspace computation of planar continuum parallel robots," *IEEE Robotics and Automation Letters*, vol. 7, no. 2, pp. 2700–2707, 2022.

[48] 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.

[49] E. and F. Cosserat, *Theorie des corps deformables*. Hermann-Paris, 1909.

[50] J. Simo, "On the dynamics in space of rods undergoing large motions - a geometrically exact approach," *Computer Methods in Applied Mechanics and Engineering*, vol. 66, no. 2, pp. 125–161, 1988.

[51] 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.

[52] M. Bergou, M. Wardetzky, S. Robinson, B. Audoly, and E. Grinspun, "Discrete elastic rods," *ACM transactions on graphics*, vol. 27, no. 3, pp. 1–12, 2008.

[53] F. Boyer, M. Porez, and W. Khalil, "Macro-continuous computed torque algorithm for a three-dimensional eel-like robot," *IEEE Trans. Robot.*, vol. 22, no. 4, pp. 763–775, 2006.

[54] M. Mahvash and P. Dupont, "Stiffness control of surgical continuum manipulators," *IEEE Trans. Robot.*, vol. 27, no. 2, pp. 334–345, 2011.

[55] F. Renda, F. Boyer, J. Dias, and L. Seneviratne, "Discrete cosserat approach for multisection soft manipulator dynamics," *IEEE Transactions on Robotics*, vol. 34, no. 6, pp. 1518–1533, 2018.

[56] J. Till, V. Aloil, and D. 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, 2019.

[57] 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.

[58] C. Armanini, I. Hussain, M. Iqbal, D. Gan, D. Prattichizzo, and F. Renda, "Discrete cosserat approach for closed-chain soft robots: Application to the fin-ray finger," *IEEE Transactions on Robotics*, vol. 37, no. 6, p. 2083–2098, 2021.

[59] C. Quennouelle and C. Gosselin, "Kinematostatic modeling of compliant parallel mechanisms," *Meccanica*, vol. 46, no. 1, pp. 155–169, jan 2011.

[60] R. M. Murray, Z. Li, and S. S. Sastry, *Robotic Manipulation*. CRC Press, 1994.

[61] S. Briot and F. Boyer, "Technical report associated with the paper: "A Geometrically-Exact Assumed Strain Modes Approach for the Geometrico- and Kinemato-static Modellings of Continuum Parallel Robots"," Laboratoire des Sciences du Numérique de Nantes (LS2N), Tech. Rep., 2022. [Online]. Available: https://hal.archives-ouvertes.fr/hal-03836288

[62] A. L. Orekhov and N. Simaan, "Solving Cosserat rod models via collocation and the Magnus expansion," in *Proceedings of the 2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2020)*, 2020, pp. 8653–8660.

[63] J. Nocedal and S. Wright, *Numerical Optimization*, 2nd ed. Springer, 2006.

[64] E. Gekeler, *Mathematical Methods for Mechanics*. Berlin Heidelberg: Springer-Verlag, 2008.

[65] D. Rucker and R. Webster, "Computing jacobians and compliance matrices for externally loaded continuum robots," in *Proceedings of the 2011 IEEE International Conference on Robotics and Automation (ICRA 2011)*, 2011, pp. 945–950.

[66] H. Ziegler, *Principles of Structural Stability*. Cambridge, MA, USA: Birkhäuser, 1977.

[67] Q. Peyron, K. Rabenorosoa, N. Andreff, and P. Renaud, "A numerical framework for the stability and cardinality analysis of concentric tube robots: Introduction and application to the follow-the-leader deployment," *Mechanism and Machine Theory*, vol. 132, pp. 176–192, 2019.

[68] H. L. J. Linn and M. Arnold, "Multi-body dynamics simulation of geometrically exact cosserat rods," *Multibody System Dynamics*, vol. 25, no. 3, pp. 285–312, 2011.

[69] 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, no. 5, pp. 1109–1122, 2014.

Sébastien Briot received the B.S. and M.S. degrees in mechanical engineering in 2004 and the Ph.D. degree in robotics (under the supervision of Prof. V. Arakelian) in 2007, from the Institut National des Sciences Appliquées de Rennes, Rennes, France. He was a Postdoctoral Fellow with the Ecole de Technologie Supérieure, Montreal, QC, Canada, in 2008. In 2009, he has been recruited at CNRS as a researcher in the Laboratoire des Sciences du Numérique de Nantes, Nantes, France, where he has been the Head of the ARMEN Research Team since 2017. Since 2022, he is CNRS Director of Research in the same lab. He has authored more than 50 referred journal papers, two books, and three inventions. His research interests include the design optimization of robots and the analysis of their performance, especially their singularities. Dr. Briot received the Best Ph.D. Thesis Award in Robotics from the French CNRS in 2007. In 2011, he received two other awards: the Award for the Best Young Researcher from the French Region Bretagne and the Award for the Best Young Researcher from the French Section of the American Society of Mechanical Engineering.

Frédéric Boyer was born in France in 1967. He received the Diploma degree 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, 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 ReV Team, Laboratoire des Sciences du Numérique de Nantes. His current research interests include structural dynamics, geometric mechanics, and biorobotics (locomotion dynamics and underwater electric sensing).

Dr. Boyer was the recipient of 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.