Coupling numerical deformable models in global and reduced coordinates for the simulation of the direct and the inverse kinematics of Soft Robots

Yinoussa Adagolodjo, Federico Renda, Christian Duriez

Coupling numerical deformable models in global and reduced coordinates for the simulation of the direct and the inverse kinematics of Soft Robots.

Yinoussa Adagolodjo[1], Federico Renda[2] and Christian Duriez[1]

Abstract—In this paper, we propose a method to combine the Finite Element Method (FEM) with Discrete Cosserat Modeling (DCM) to capture the mechanics and the actuation of soft robots. The FEM is used to simulate the non-linear behavior of the volume of the soft structure while the cable/rod used for the actuation is modeled using the DCM. The two models are linked using kinematic constraints without imposing meshing rules. We demonstrate that both direct and inverse kinematic models can be obtained by quadratic optimization. The originality of this coupling is that the FEM model uses global coordinates (the position of the nodes of its mesh in space) where the Cosserat model uses local coordinates (successive strain values). The coupling of these mechanical models allows to combine the best of each parametrization. On the one hand, FEM allows to capture the behavior of the volume structure of the robot while accounting for its geometry with a complex mesh. On the other hand, the DCM allows efficient modeling of 1D structures such as rods, (concentric) tubes, cables, etc. that are used to deform the volume structure of the soft robots. DCM handles large deformation, torsion and (in)-extensibility and is efficient to compute. Moreover, the approach is compatible with complementarity constraints introduced when modeling contact and friction of the robot with its environment as well as the self-collision.

Index Terms—Flexible Robotics, Soft Robot Applications, Contact Modeling, Grasping, Modeling, Simulation and Animation, Tendon/Wire Mechanism, Control, and Learning for Soft Robots

I. INTRODUCTION

THE field of soft robots opens the way to many new design possibilities, such as the choice of actuators, the transmission of forces in the structure, the use of materials with different stiffness, etc. On the one hand, the possible combinations to obtain desired motions on a deformable robot are multiple. On the other hand, the tools for the designer, still require improvement, especially when it comes to mechanical modeling.

So far, two methods have mainly emerged to succeed in modeling the mechanics of these robots, while taking into account their material properties: The first one is based on the Cosserat theory, which has the advantage of having a formulation close to serial rigid robots using reduced coordinates. The second group's numerical models of finite element type, which generate large models whose state is defined in global coordinates.

A. Motivations to combine Cosserat rods and FEM

The DCM is totally adapted to curved robots with a constant or quasi-constant cross-section and the reduced coordinates make it possible to limit the size of the models and therefore calculation times are fast. Furthermore, if one wishes to particularly constrain a dimension (for example, to simulate the inextensible behavior of a rod or cable), it is sufficient to not introduce this strain in the reduced coordinates of the Cosserat model. However, this model is not adapted for the simulation of truss structures, complex geometries and cannot be used for volume deformations.

Conversely, the state of a FEM model is often represented by the position and velocity (in absolute coordinates) of a set of nodes connected by elements. This leaves a very large freedom in geometry and allows to establish well-defined boundary conditions on these numerical models. There are FEM models for rods (beams), for shells, or for modeling deformable volume solids. Material laws can be defined very freely and for volume models, anisotropy can be modeled in all directions and rods can be connected freely to create truss structures. However, for rods modeling, it is necessary to add additional constraints (such as Lagrange multipliers) to really avoid some deformations, such as in-extension and this tends to deteriorate the numerical conditioning of the model. Here, the DCM is more suitable.

One can therefore observe that each of the models has advantages for modeling the deformations of certain types of solids. Yet, there is an increasing interest in combining different types of bodies in the design of soft robots. There are recurrent cases where it would be useful to combine these models: one of them is to model the stiffness of cables used to transmit motion to a soft robot with a deformable volumetric structure as demonstrated in this paper. Further examples are the use of an inextensible wire placed around a pneumatic cavity to limit its radial inflation or to design new robots that would use pre-bent flexible rods to transmit their motion to a volumetric structure. These are possible extensions of this work.

B. Related Work

The cosserat model has been introduced in continuum robotics to simulate the deformation the robot body whose geometry and mechanical characteristics are similar to a rod. By extension, this model can be used to simulate needles and sutures [1] or wires [2]. The specificity of Cosserat's theory from the point of view of the mechanics of continuous media is to consider that the set of material points consists of rigid solids (3 translations, 3 rotations), where most other models of continuum media mechanics consider the material points as particles (3 translations). For the modeling of linear structures, it is therefore possible to find a framework very close to the articulated solids with a series of rigid solids whose relative position is defined by a strain state. This model can be used to model and control concentric tube robots [3], continuum robots actuated with cables [4] or pneumatic soft robots with a constant cross-section [5] There are several formalism to model Cosserat rods. In this paper, we have developed the Piecewise Constant Strain (PCS) presented by [6], [7] in the SOFA framework.

The use of finite element models for the design of robots and especially soft robots is widely used [8], [9], [10], [11]. Indeed, finite element methods combined with suitable constitutive laws provide a valuable tool for simulating the behavior of these robots in interaction with their environment. However, its use for real-time modeling and control is less common. Yet, several approaches allow for very fast computations but sometimes at the price of limitations in the constitutive laws or in the extent of deformation (very large deformations). For instance, the co-rotational technique [12], [13], which is used to improve the computation time of the FEM model, is limited to small-strain (but large displacements and rotations). In [14], a model order reduction approach is proposed to reduce the size of the model by introducing another parameterization based on deformation modes.

FEM beam models allow to model rods like the Cosserat model but each node has 6 degrees of freedom, defined in global coordinates. The model is therefore often much larger than a corresponding Cosserat model. Inversely, truss structures can easily be simulated with these beams [9], [15], which is very difficult with most of the Cosserat rod models.

These two approaches have in common that they derive from the continuous media mechanics. It is obvious that each of them has its own advantages and disadvantages. It is useless to try to place one above the other. On the contrary, to benefit from their respective advantages, it is much more useful to combine them. In our opinion, their respective advantages are mainly related to the parameterization choices (reduced and global).

There is obviously a lot of work on this subject for rigid articulated models, for instance recently in [16], but surprisingly very little for deformable models, especially in robotics.

C. Contribution

To our best knowledge, we propose the first study of a mixed modeling for soft robots, which uses a deformable model in global coordinates (FEM type) to model the structure of the robot and a deformable model in reduced coordinates (Cosserat rod types) to model the tendons passing through this structure to deform it. This work, which is relatively pioneering for deformable models, is part of the process of analyzing the parameterization choices that already exist for articulated rigid bodies, even if there are important differences.

II. MECHANICAL MODELS

In [17], a general formulation of the kinematic model of soft robots controlled in position has been proposed:

$$\Delta\boldsymbol{\delta}_{\mathrm{e}} = \mathbf{W}_{\mathrm{ea}}\mathbf{W}_{\mathrm{aa}}^{-1}\Delta\boldsymbol{\delta}_{\mathrm{a}} \tag{1}$$

In this equation $\Delta\boldsymbol{\delta}_{\mathrm{e}}$ is a *small motion* of the effector(s), $\Delta\boldsymbol{\delta}_{\mathrm{a}}$ a *small motion* of the actuator(s) and $\mathbf{W}_{\mathrm{ea}}$ and $\mathbf{W}_{\mathrm{aa}}$ are the compliance matrices (inverse of stiffness) that relates respectively, forces in the actuator space to displacements in the effector space and forces and displacements in the actuator space. This equation is obtained by linearization of a numerical model of the deformations. By *small motion*, we mean motions that are small enough over a time step that the assumption of a linear motion is correct (like Jacobian of an articulated system in a kinematic equation of a rigid robot) The $\mathbf{W}_{ij}$ compliance matrices are updated at each simulation step, using a mechanical model that accounts for the mechanical properties of the materials and the non-linearities of the deformations.

In this section we explain how we compute these using both DCM and FEM model. Moreover, the above formulation supposes that both actuators and effectors are defined on a unique deformable model so that $\mathbf{W}_{ij} \neq \mathbf{0}$ which is not the case in our study where actuators are defined on Cosserat rods and effectors on the FEM model. We will explain how we integrate sliding constraints in the model and update the kinematics in equation (14) for coupled models.

A. Discrete Cosserat Model (DCM)

In this work, the state of the DCM model, based on Hook's law, is defined by a 6DOF frame at the base and a series of piecewise constant strain which can be assimilated as reduced coordinates defined between rigid bodies.

1) Parametrization and Kinematics: In Cosserat theory, each material point i is defined as a rigid body (with a position $\mathbf{u_i}$ and a rotation $\mathbf{R}_i$ $g_0(X = 0)$)

$$g_i(X) = \begin{pmatrix} \mathbf{R}_i & \mathbf{u_i} \\ \mathbf{0}^T & 1 \end{pmatrix} \tag{2}$$

with $g_i(X) \in \mathrm{SE}(3)$. This material point i belongs to a curve can be defined with respects to its predecessor. Derivatives can then be defined along the curve to extract the strain state (see [6] for details). $g_i(\cdot) = X \in [0, L_i] \mapsto g_i(X) \in \mathrm{SE}(3)$ (respectively by a point)

$$g_i' = g_i\,\hat{\xi}_i \tag{3}$$

where $\hat{\xi}_i(X) = \begin{pmatrix} \tilde{\mathbf{k}}_i & \mathbf{x} \\ \mathbf{0}^T & 0 \end{pmatrix} \in \mathfrak{se}(3)$, $\xi(X) = (\mathbf{k}_i^T, \mathbf{u_i}^T)^T \in \mathbb{R}^6$, defines the strain state (with $\tilde{\mathbf{k}}_i(X) \in \mathfrak{so}(\mathbf{3})$, $\mathbf{k}_i(X) \in \mathbb{R}^3$

and $\mathbf{x}(X) \in \mathbb{R}^3$ respectively the angular and linear strain). The model is parametrized using this strain to trigger the deformation directions and to be as close as possible to the behavior law of the material (defined from the strain). To model constrained rod, such as the Kirchhoff-Love case with angular strain only, the strain field is specified as: $\xi_i(X) = \mathbf{B}_i \mathbf{q}_i + \bar{\xi}_i$, where $\mathbf{B}_i \in \mathbb{R}^{6 \times n_i}$ forms a basis for the allowed motion subspace, $\mathbf{q}_i(X) \in \mathbb{R}^{n_i}$ contains the values of the allowed strains and, $\bar{\xi}_i \in \mathbb{R}^6$ is a fixed twist modeling constant yet non-zero strains (e.g., for in-extensible rod with x-axis tangent to the center-line $\bar{\xi}_i = [000100]^T$).

Now, assuming piece-wise constant strains [6] , equations (1) can be analytically integrated using the matrix exponential method, leading to :

$$g_i(X) = e^{X \hat{\xi}_i} \qquad (4)$$

In the same way, thanks to Lie derivatives toolbox, one can write the velocity twists $\eta_i(X)$ of a rigid material points i using the one of its predecessor and the temporal derivative of the allowed strains $\dot{\mathbf{q}}_i$ (velocities of reduced coordinates). See [6] for details.

Successive applications of the kinematics for all the discrete body points of the system along the curve, yields to the definition of the geometric Jacobian $\mathbf{J}(X) \in \mathbb{R}^{6 \times n}$ and its derivative $\dot{\mathbf{J}}(X) \in \mathbb{R}_{6 \times n}$ (n being the total number of DOFs) for each soft/rigid body, which relates the generalized coordinate vector $\mathbf{q} = [\mathbf{q}_0^T \mathbf{q}_1^T \dots \mathbf{q}_{N-1}^T] \in \mathbb{R}^n$ (N being the total number of bodies) and the ith velocity twist as shown below.

$$\eta_i(X) = \sum_{h=0}^{i} \mathbf{Ad}_{g_h \dots g_i}^{-1} \mathbf{T}_{g_h} \mathbf{B}_h \dot{\mathbf{q}}_h = \sum_{h=0}^{i} {}^i \mathbf{S}_h \dot{\mathbf{q}}_h = \mathbf{J}_i \dot{\mathbf{q}}$$
$$\dot{\eta}_i(X) = \sum_{h=0}^{i} {}^i \mathbf{S}_h \ddot{\mathbf{q}}_h + {}^i \dot{\mathbf{S}}_h \dot{\mathbf{q}}_h = \mathbf{J}_i \ddot{\mathbf{q}} + \dot{\mathbf{J}}_i \dot{\mathbf{q}} \qquad (5)$$

where the block elements of the ith Jacobian ${}^i \mathbf{S}_{(.)} \in \mathbb{R}^{6 \times n_{(.)}}$ and its derivative ${}^i \dot{\mathbf{S}}_{(.)} \in \mathbb{R}^{6 \times n_{(.)}}$ have been defined. Note that to simplify the notation, we will consider in the following that the 6 DOF movement of the rigid body motion $\mathbf{q}_0$ of the base belongs to the state vector $\mathbf{q}$.

The vector of the generalized coordinates being described in a reduced space of strains, we can write the behavior law of the material in this space to obtain the stresses σ_i created by the deformations. This formulation therefore has the advantage of having a formulation of internal forces that is extremely simple to calculate, with diagonal block matrices linking q_i to σ_i. These stresses must be balanced with the forces applied to the rod (especially gravity, the inertial forces, and also the forces used to couple with the FEM). Exactly, as for rigid articulated chains, the transposed Jacobian matrices $\mathbf{J}_i^T$ are used to bring back the forces defined on the rigid material points to the reduced coordinate space (see schema at Fig.1). We can therefore write the equation of the dynamics in the reduced space:

$$\mathcal{M}(\mathbf{q}) \ddot{\mathbf{q}} + \mathcal{C}(\mathbf{q}, \dot{\mathbf{q}}) \dot{\mathbf{q}} = \mathbf{J}^T \boldsymbol{P} - \mathcal{F}(\mathbf{q}_i) + \mathbf{J}^T \boldsymbol{\lambda} \qquad (6)$$

with $\mathcal{M}(\mathbf{q})$ the projection of the mass matrix in the reduced space, $\mathcal{C}(\mathbf{q}, \dot{\mathbf{q}}) \dot{\mathbf{q}}$ some derivative terms that are similar to

centrifugal and Coriolis forces. $\mathbf{J}^T \boldsymbol{P}$ and $\mathbf{J}^T \boldsymbol{\lambda}$ are the gravitational forces and the interaction forces projected in the reduced space and $\mathcal{F}(\mathbf{q}_i)$ represents the constitutive law.

B. FEM in global coordinates

To simulate the behavior of soft robots with volume deformations in interaction with their environment, we made the choice of the FE methods computed in real-time. This part is mainly based on the previous works developed in [10], [18], [11], which are using FEM implementation provided by the simulation framework SOFA [19]. In this paper, the choice of the co-rotational model is made for FE method, which a good trade of between computational efficiency and accuracy of the model. Indeed, although a linear model, the co-relational model allows to take into account the geometric non-linearity due to the rotation of the elements during the deformation. Which is the main limitation of most of linear models in FEM.

Tetrahedral or hexahedral meshes are used for the representation of the robot body in the FEM implementation for deformation of the volume structure. In order to efficiently connect these complex topologies, global coordinates are used : the state of the system is represented by the motions (position and velocity) of the nodes of the mesh which are linked together by the integrated constitutive law on each of the elements. We are therefore in the opposite case to Cosserat's model: the stress/strain space in which we define the behavior of the material is a *mapped* space.

On the other hand, one can easily describe in nodes space the forces related to mass (gravity, inertia), the coupling constraints with the Cosserat model or the contact constraints. Newton's second law that describes the dynamic equation of the deformable structure of the soft robot is therefore written in absolute coordinates:

$$\mathbf{M} \ddot{\mathbf{x}} = \boldsymbol{P} - \mathcal{F}(\mathbf{x}, \dot{\mathbf{x}}) + \mathbf{H}^T \boldsymbol{\lambda} \qquad (7)$$

where $\mathcal{F}$ is a non-linear function providing the internal forces of the deformable structure (here the soft robot) computed by the FEM. $\mathbf{M}$ and $\mathbf{x}$ are respectively the mass matrix (constant and diagonal matrix when using mass lumping) and
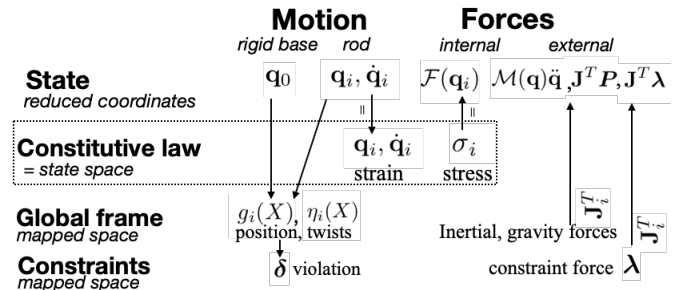
Fig. 1. In the Cosserat model, the state of the system is calculated in reduced coordinates in the space of the material constitutive law. The global positions are deduced by integrating the motion along the curve. Similarly, the forces defined in the global reference frame (gravity, inertia, constraints, etc.) are transferred to the reduced space. The constraint space (to interact with the FEM), is also using positions defined in the global frame so, it requires the same process to compute violation and transfer constraint force.

the vector of the nodes position given by the FE method. $\boldsymbol{P}$ represents the known external forces (such as gravity), $\boldsymbol{\lambda}$ and $\mathbf{H}$ are respectively the vector of Lagrange multipliers and the matrix containing the constraint directions.

C. Definition of Constraints

To make the complete modeling of the robot and in particular the coupling between the models, the taking into account of the contacts and friction or the definition of the effector or the directions of actuation, we rely on the use of Lagrange multipliers, associated to several type of constraints. These constraints are defined in their own spaces, often global coordinates (to define a contact or sliding direction). As shown above, it is important to be able to easily transfer constraints from a constraint space to the state space (reduced for the Cosserat, global using nodes for the FEM.See Fig.1 and Fig.2 for a better overview). This easy transfer of constraints through different spaces is facilitated by the concept of mappings developed in SOFA [19].

In the following of this section, we will present how the Lagrange Multipliers is used for modeling the sliding actuation of rods, handle contacts with the external environment and to guide the robot's end effector to reach a define target.

*1) **Cable Actuation**:* In our study, the soft robot is actuated using cables or rods, and the loads applied by these actuators is model using the Lagrange's multipliers to transmit the forces on the FEM to deform it. The coupling is bilateral: while deforming, the FEM will also cause the deformation of the Cosserat model by imposing sliding link forces.

As mentioned above, the cable is modeled using discrete Cosserat modeling and divided into N sections of the form $[0, L_1), (L_1, L_2)(L_{N-1}, L_N]$ (with $L_N = L$). At any time t of the simulation, one can obtain the rigid material point $g_i(X)$ on the cable at the abscissa X. We assume that the cable is attached to the robot at its extremity $P_{fix} = g_i(L_N)$, and we actuated the robot by pulling the rigid body at the base of the

Cosserat model $P_{pull} = \mathbf{q}_0$. The pulling translation $\boldsymbol{\delta}_a(\mathbf{q})$ of the cable at the base deforms the robot, and create a reaction force on this base that is computed thanks to the Lagrange multiplier $\boldsymbol{\lambda}_a$ (This corresponds to the force that a sliding actuator should apply to create the translation). In the model, this rigid body base is fixed in all directions (translations and rotations) except the one in which the cable is pulled.

Note that we have also used the same formulation to model a torsion based actuation. In such case $\boldsymbol{\delta}_a(\mathbf{q})$ is an angular displacement imposed along the main axis of the rod.
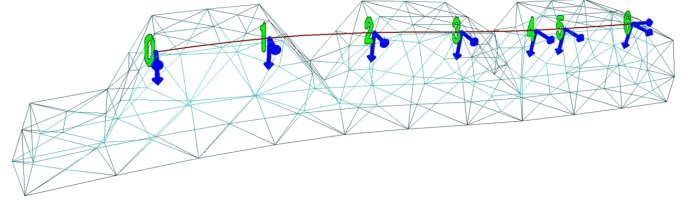
Fig. 3. Coupling between the DCM model (in red) and the FEM model (volume mesh) using sliding constraints (numbered from 0 to 5). The last constraint (6) has an additional direction to reproduce the fixation of the cable extremity in the silicone.

*2) **Sliding Constraint**:* To model the coupling forces, $\mathbf{H}_s^T \boldsymbol{\lambda}_s$, between the FEM and the Cosserat models, we first defined constraint points P_s (3D points , see fig3), inside the FEM mesh. These points are chosen very close to the path of the cable inside the robot structure. The first point is positioned on the entry point of the cable in the robot and the last at the fixing point P_{fix}, of the cable on the robot. The rest of the points are homogeneously distributed along the cable (see fig. 4).

At each time step, for each of these points, we compute the projection on the cable given the current configuration (see green line in fig. 4). We can then define a 2d vector $\boldsymbol{\delta}_s$ between the point and its projection, in the normal plane of the cable's curve (see blue arrows in fig 4). For the last point, which corresponds to the extremity of the cable $P_{fix} = g_i(L_N)$, we use a 3d vector $\boldsymbol{\delta}_s$ that measure the distance between the point in the FEM and this extremity. Thanks to the mapping between spaces defined above (see Figs 1 and 2) this vector can be linked to the position state of both models $\boldsymbol{\delta}_s(\mathbf{q}, \mathbf{x})$.

The role of the associated Lagrange Multiplier $\boldsymbol{\lambda}_s$ will be to force $\boldsymbol{\delta}_s = 0$ at the end of each step (for more details about bilateral constraints, see [10]). Jacobian matrices are both defined on the Cosserat model $\mathbf{J}_{0,s} = \frac{\partial \boldsymbol{\delta}_s}{\partial \mathbf{q}_0}$, $\mathbf{J}_{i,s} = \frac{\partial \boldsymbol{\delta}_s}{\partial \mathbf{q}_i}$ and on the FEM $\mathbf{H}_s = \frac{\partial \boldsymbol{\delta}_s}{\partial \mathbf{x}}$ to transfer the constraint force in the state space of each model: $\mathbf{J}_{0,s}^T \boldsymbol{\lambda}_s$ $\mathbf{J}_{i,s}^T \boldsymbol{\lambda}_s$ and $\mathbf{H}_s^T \boldsymbol{\lambda}_s$

*3) **Effector Constraint**:* Our objectives in most of the simulations, direct or inverse, that we conduct is to guide the robot extremity called effector $\mathbf{e}$, towards a target in its workspace. This point belongs to the FEM mesh and its motion can be interpolated using node motions and interpolation
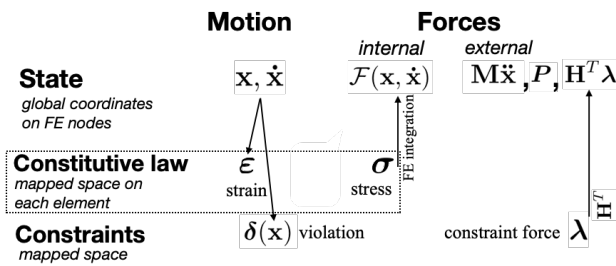

Fig. 2. In the case of the FEM model, the state of the system is in global coordinates with node positions and derivatives (this is convenient for gravity $\boldsymbol{P}$ and dynamic forces $\mathbf{M\ddot{x}}$). On the other hand, the calculation of internal forces $\mathcal{F}(\mathbf{x}, \dot{\mathbf{x}})$ requires a change from this global state to the space of the constitutive law (strain / stress). By integrating the stress on the volume of each element, we obtain the corresponding internal forces on each node. Finally, interaction constraints $\mathbf{H}^T \boldsymbol{\lambda}$ are potentially defined in another space to be able to place the constraints between nodes of the mesh. $\boldsymbol{\delta}(\mathbf{x})$ represents the violation of the constraint (most of the time a distance) of any interpolated point $\mathbf{p}$ in the mesh. $\mathbf{H} = \frac{\partial \boldsymbol{\delta}}{\partial \mathbf{x}}$ can also be seen as the Jacobian of the mapping between the FE space and the constraint space. Note that to link the motion of $\mathbf{p}$ and the state $\mathbf{x}$, we use FEM interpolation functions $\mathbf{p} = \Psi(\mathbf{x})$.
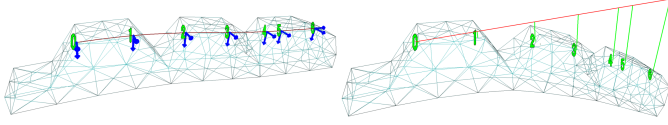
Fig. 4. Sliding constraints, applied on constraint points during soft robot actuation. Blue arrows are bilateral constraints, used to maintain constraint points on the cable. Green line are violations.

functions Ψ. Let's define $\boldsymbol{\delta}_e(\mathbf{x}) = \Psi(\mathbf{x}) - \mathbf{u}_d$ as the violation function that measures the shift along x, y and z between this point and the desired position $\mathbf{u}_d$. Like above we compute the Jacobian matrix, $\mathbf{H}_e$ of this vector $\boldsymbol{\delta}_e(\mathbf{x})$. Since in our various simulations, we will not have to apply a load to the robot's effector, we can therefore note $\boldsymbol{\lambda}_e = \mathbf{0}$.

4) Friction Contacts: To manage the different contacts that can appear between the robot and its environment, we combine Signorini's law for contact and Coulomb's law for friction (see [10] for more details). The force $\mathbf{H}_c^T \boldsymbol{\lambda}_c$ induced by this contact is divided into normal $\mathbf{H}_n^T \boldsymbol{\lambda}_{nc}$ and tangential $\mathbf{H}_t^T \boldsymbol{\lambda}_{tc}$ forces. The corresponding matrix of directions $\mathbf{H} = [\mathbf{H}_n \ \mathbf{H}_t]^T$ holds the normal and the tangential(friction) direction of the contact force, and $\boldsymbol{\lambda}_c$ is the contact response. The contact appears on the FEM structure and the violation function $\boldsymbol{\delta}_c(\mathbf{x})$ measures at the contact point, the shift between the robot and the obstacle. When a potential contact with the robot has been detected, we need to solve the contact response $\boldsymbol{\lambda}_c$ at the collision point with complementarity conditions.

III. DIRECT AND INVERSE SOLVING

To integrate in time, we made the choice of an implicit backward Euler scheme. This integration scheme is known to be dissipative but it guarantees unconditional stability and being a low-order scheme, non-smooth events such as contacts can be integrated using a time-stepping approach. Using this integration scheme for (6) and (7), we obtain non-linear problems.

Instead of directly solving these nonlinear problems at each simulation step, we solve a linearized problem. Thus, we can compute a linearization of the $\mathcal{F}$ by applying a first order Taylor series expansion. For instance, on FEM:

$$\mathcal{F}(\mathbf{x}_h, \dot{\mathbf{x}}_h) = \mathcal{F}(\mathbf{x}_0, \dot{\mathbf{x}}_0) + \frac{\partial \mathcal{F}}{\partial \mathbf{x}} d\mathbf{x} + \frac{\partial \mathcal{F}}{\partial \dot{\mathbf{x}}} d\dot{\mathbf{x}} \qquad (8)$$

with $\mathbf{x}_h$ and $\mathbf{x}_0$ the position of the nodes respectively at the end and at the beginning of the step.

We do the same for the internal forces in the Cosserat model but in this work, we use a linear constitutive law (Hooke's law) so the function $\mathcal{F}(\mathbf{q}_i)$ is already linear so $\frac{\partial \mathcal{F}}{\partial \mathbf{q}}$ is constant. In the Cosserat model the Mass $\mathcal{M}(\mathbf{q})$ is non linear but we take its value at the beginning of the step $\mathbf{q}_0$, same for $\mathcal{C}(\mathbf{q}, \dot{\mathbf{q}})$

Let's consider the simulation at time $t_h = t_0 + h$ using the relations $d\mathbf{q} = \mathbf{q}_h - \mathbf{q}_0 = h\dot{\mathbf{q}}_h$ and $d\mathbf{x} = \mathbf{x}_h - \mathbf{x}_0 = h\dot{\mathbf{x}}_h$ to integrate positions and $d\dot{\mathbf{q}} = \dot{\mathbf{q}}_h - \dot{\mathbf{q}}_0 = h\ddot{\mathbf{q}}_h$ and $d\dot{\mathbf{x}} = \dot{\mathbf{x}}_h - \dot{\mathbf{x}}_0 = h\ddot{\mathbf{x}}_h$ for the velocities, we have:

$$\underbrace{\left(\mathcal{M}(\mathbf{q}_0) + h\mathcal{C}(\mathbf{q}_0, \dot{\mathbf{q}}_0) + h^2 \frac{\partial \mathcal{F}}{\partial \mathbf{q}} \right)}_{\mathbf{A_q}} \underbrace{d\dot{\mathbf{q}}}_{\mathbf{u_q}} = h^2 \frac{\partial \mathcal{F}}{\partial \mathbf{q}} \dot{\mathbf{q}}_0 \qquad (9)$$
$$+ h(\mathbf{J}^T \mathbf{P} - \mathcal{F}_0) + h\mathbf{J}^T \boldsymbol{\lambda}$$

for the Cosserat model and for the FEM model:

$$\underbrace{\left(\mathbf{M} + h\frac{\partial \mathcal{F}}{\partial \dot{\mathbf{x}}} + h^2 \frac{\partial \mathcal{F}}{\partial \mathbf{x}} \right)}_{\mathbf{A_x}} \underbrace{d\dot{\mathbf{x}}}_{\mathbf{u_x}} = h^2 \frac{\partial \mathcal{F}}{\partial \mathbf{x}} \dot{\mathbf{x}}_0 \qquad (10)$$
$$+ h(\boldsymbol{P}_h - \mathcal{F}_0) + h\mathbf{H}^T \boldsymbol{\lambda}$$

with $\boldsymbol{P}_h = P(t_h)$ and $\mathcal{F}_0 = \mathcal{F}(\mathbf{x}_0, \dot{\mathbf{x}}_0)$.

These two matrix systems of equations are coupled with the values of the constraint $\boldsymbol{\lambda}$. As described in [19], we first compute each system independently by setting $\boldsymbol{\lambda} = \mathbf{0}$ and obtain the free motion.

A. Direct Problem Formulation

In a direct problem formulation, the displacement of the actuator $\boldsymbol{\delta}_a(\mathbf{q}) = \mathbf{v}$ is known. Moreover, we know that the sliding constraints are solved when $\boldsymbol{\delta}_s(\mathbf{q}, \mathbf{x}) = \mathbf{0}$.

These constraints depend on the integrated equations of DCM (9) and (10). After solving the free motion, we can compute the Schur complement for each constraint:

$$\boldsymbol{\delta}_i = \underbrace{h^2 [\mathbf{J}_i \mathbf{A_q}^{-1} \mathbf{J}_j^T + \mathbf{H}_i \mathbf{A_x}^{-1} \mathbf{H}_j^T]}_{\mathbf{W}_{ij}} \boldsymbol{\lambda}_j + \boldsymbol{\delta}_j^{\text{free}} \qquad (11)$$

Where $\boldsymbol{\delta}_j^{\text{free}} = \boldsymbol{\delta}_j(\mathbf{q}_0, \mathbf{x}_0) + \mathbf{J}_j d\mathbf{q}^{\text{free}} + \mathbf{H}_j d\mathbf{x}^{\text{free}}$.

The physical meaning of this Schur complement is central in the method. $\mathbf{W}_{ij}$ provides a measure of the instantaneous mechanical coupling between the boundary conditions i and j, whether they correspond to actuator or sliding. In practice, this projection allows to perform the optimization with the smallest possible number of equations.

It should be emphasized that one of the main difficulties is to compute $\mathbf{W}_{ij}$ in a fast manner, in particular for the FEM model because the size of $\mathbf{A_x}$ is large, whereas $\mathbf{A_q}$ is much smaller. No precomputation is possible because the value changes at each iteration. But this type of projection problem is frequent when solving friction contact on deformable objects, thus several strategies are already implemented in SOFA.

For solving the direct model we have to solve this linear system to obtain $\boldsymbol{\lambda}_a$ and $\boldsymbol{\lambda}_s$

$$\begin{bmatrix} \boldsymbol{\delta}_a = \mathbf{v} \\ \boldsymbol{\delta}_s = \mathbf{0} \end{bmatrix} = \begin{bmatrix} \mathbf{W}_{aa} & \mathbf{W}_{as} \\ \mathbf{W}_{sa} & \mathbf{W}_{ss} \end{bmatrix} \begin{bmatrix} \boldsymbol{\lambda}_a \\ \boldsymbol{\lambda}_s \end{bmatrix} + \begin{bmatrix} \boldsymbol{\delta}_a^{\text{free}} \\ \boldsymbol{\delta}_s^{\text{free}} \end{bmatrix} \qquad (12)$$

If we add friction contact, we obtain a Mixed Complementarity Problem that we can solve with a Gauss-Seidel algorithm implemented in SOFA [10]. When $\boldsymbol{\lambda}_a$ and $\boldsymbol{\lambda}_s$ are available, equations (9) and (10) are evaluated again to obtain the final motion.

B. Kinematic and Inverse Problem Formulation

To compute the kinematics and the inverse problem formulation, we add the constraint that corresponds to the effector. We can rewrite the Schur complement:

$$\begin{bmatrix} \boldsymbol{\delta}_e \\ \boldsymbol{\delta}_a \\ \boldsymbol{\delta}_s = 0 \end{bmatrix} = \begin{bmatrix} \mathbf{W}_{ee} & \mathbf{W}_{ea} & \mathbf{W}_{es} \\ \mathbf{W}_{ae} & \mathbf{W}_{aa} & \mathbf{W}_{as} \\ \mathbf{W}_{se} & \mathbf{W}_{sa} & \mathbf{W}_{ss} \end{bmatrix} \begin{bmatrix} \mathbf{0} \\ \boldsymbol{\lambda}_a \\ \boldsymbol{\lambda}_s \end{bmatrix} + \begin{bmatrix} \boldsymbol{\delta}_e^{\text{free}} \\ \boldsymbol{\delta}_a^{\text{free}} \\ \boldsymbol{\delta}_s^{\text{free}} \end{bmatrix} \quad (13)$$

We can re-write the kinematics of a soft robot, previously described at equation (1), to provide the link between the motion of the effector and the actuator on two different models coupled by sliding constraints:

$$\Delta\boldsymbol{\delta}_e = (\mathbf{W}_{ea} - \mathbf{W}_{es}\mathbf{W}_{ss}^{-1}\mathbf{W}_{sa})\mathbf{W}_{aa}^{-1}\Delta\boldsymbol{\delta}_a \quad (14)$$

Note that $\mathbf{W}_{ea} = 0$ as $\boldsymbol{\delta}_e$ is defined on the FEM whereas $\boldsymbol{\delta}_a$ on the DCM.

For the inverse problem, the main difference is that here both $\boldsymbol{\delta}_e$ and $\boldsymbol{\delta}_a$ are unknown. The goal is to find the values of $\boldsymbol{\lambda}_a$ (and consequently $\boldsymbol{\lambda}_s$) in order to minimize $\|\boldsymbol{\delta}_e\|^2$.

we obtain the following QP minimization problem:

$$\min_{\boldsymbol{\lambda}_a,\boldsymbol{\lambda}_s} \left\| \mathbf{W}_{ea}\boldsymbol{\lambda}_a + \mathbf{W}_{es}\boldsymbol{\lambda}_s + \boldsymbol{\delta}_e^{\text{free}} \right\|^2$$
$$with: \quad (15)$$
$$\mathbf{W}_{sa}\boldsymbol{\lambda}_a + \mathbf{W}_{ss}\boldsymbol{\lambda}_s + \boldsymbol{\delta}_s^{\text{free}} = \mathbf{0}$$

Note that we can force the use of pulling force by adding $\boldsymbol{\lambda}_a \geq \mathbf{0}$. $\boldsymbol{\lambda}_s$ is part of the optimization variables but one can use $\boldsymbol{\lambda}_s = -\mathbf{W}_{ss}^{-1}(\mathbf{W}_{sa}\boldsymbol{\lambda}_a + \boldsymbol{\delta}_s^{\text{free}})$ to replace $\boldsymbol{\lambda}_s$ in the minimization problem and only find $\boldsymbol{\lambda}_a$. But most of the QP solvers are optimized for that and avoid the costly computation of $\mathbf{W}_{ss}^{-1}$. (15) is solved using the qpOASES library [20].

Finally, computation of contact forces $\boldsymbol{\lambda}_c$ could be added as part of the optimization variables, which would allow the controller to make use of contact forces to achieve the desired motion [11]. But as contacts are complementarity constraints, one need to use a QPCC solver.

IV. RESULTS

In this paper we have presented a new modeling methodology which combines FEM and DCM to describe the behavior of deformable robots in their environment in real-time. We also presented the constraint-based kinematics of the robots, inspired by our previous work, whether in direct or inverse.

In this section we will evaluate these methods, by presenting various simulation results for robots, with different geometric and mechanical properties, and with different actuation directions. We will also associate to this paper a video that highlights each part of the proposed method and facilitate the understanding of the paper.

A. Direct Simulation: example of a Grasper

In this first part of the simulation, we use an underactuated grasper made of silicone designed by Hussan et al. in [21] then modeled in [10]. But in [10], the model of the cable is only a geometric constraint. The mechanical properties of the cable

are not taken into account. Here, the three cables are simulated with Discrete Cosserat Model. In order to allow the grasper to manipulate a cube, modeled as a rigid, in its working space, we have added friction and contact constraints to the actuation constraints. As shown in Fig.5.
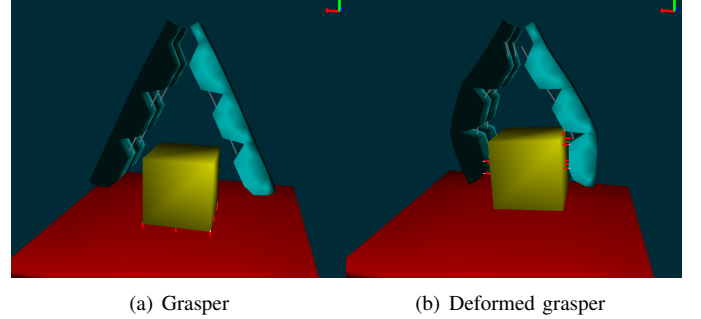

(a) Grasper   (b) Deformed grasper

Fig. 5. Direct simulation of a soft gripper that grasps a rigid object. The benefit of having a mechanical model of cables is to take into account their rigidity in the global mechanical behavior. We can also obtain the stress levels in the cables.

(a) Three section cable   (b) Six sections cable

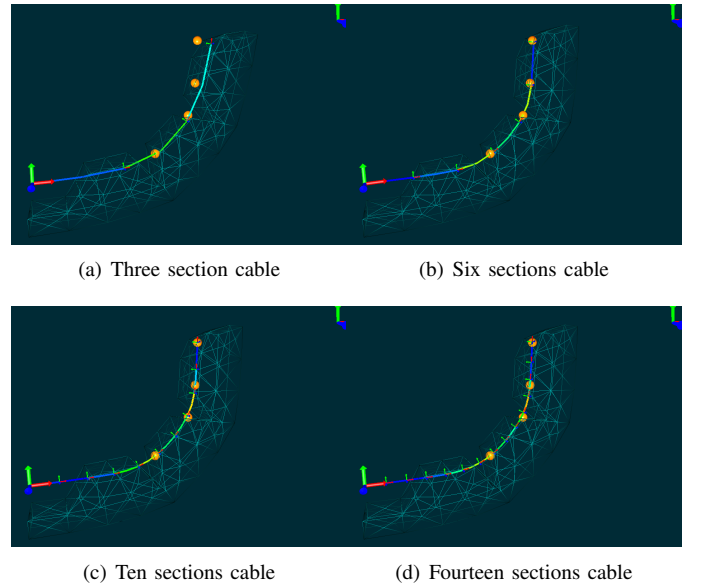(c) Ten sections cable   (d) Fourteen sections cable

Fig. 6. The study of the convergence of the model in relation to the number of sections that define the 1D object.

One of the strongest assumptions of the DCM model is that the strain is constant along each section that compose the 1D object. We thus evaluated the convergence of the model compared to the number of sections which define a cable. To do this, we first define the cable that controls the finger using 80 sections. Then the finger is deformed by pulling the cable to a given position. Four different targets are then defined on the cable (see yellow spheres in Fig.6). Then, we varied the number of sections that define the cable starting from three. Quickly, one can notice that the cable goes through the four predefined targets (after the deformation) with only six sections(see Fig.6). This guarantees the convergence of the model with respect to the number of sections used to define the object.

The other information which is represented in Fig.6 is the distribution of the bending (with respect to the Z axis) along the cable during the deformation. This goes from blue (no bending) to red (large bending). Again, one can notice that from six sections the shape of the color map is quite similar even if the intensity can vary slightly. However, it is important to remember that, the variation interval of the color map depends on the value of the simulation parameters (rod and deformable object physical properties).

An interesting aspect of using DCM to model the cable instead of using a geometric constraint is that, when we increase the radius of the section of the cable the deformation of the FEM is different for the same actuation force (see Fig.7). Indeed, in such case, the bending forces imposed by the FEM model are not sufficient for the rigidity of the rod.

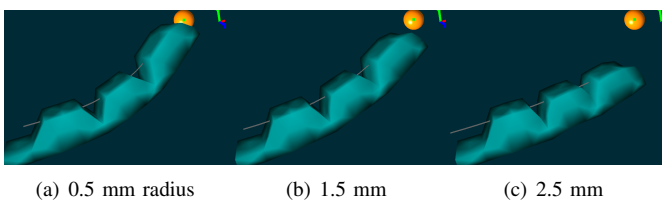(a) 0.5 mm radius   (b) 1.5 mm   (c) 2.5 mm

Fig. 7. The kinematics of the flexible finger is influenced by the properties of the cable (or rod). The larger its radius becomes, the less it can bend and the silicone tends to compress rather than bend.

Another type of test we have carried out to highlight this work is the use of pre-bent rods (as for concentric tube robots). Here, we can slide such a rod into the FEM model and when we translate or bend the rod, we observe that the FEM model deforms due to the rigidity of the rod and its pre-bent shape (see Fig.8). This type of design could be particularly interesting in future work for manipulation. Fig.8.(d) shows the distribution of the torsion along the cable which corresponds to the type of deformation observed.

B. Inverse Simulation

Inverse simulation was tested on two examples: the soft finger already presented above and a small tentacle actuated by 4 cables presented in [11]. Each time, the inverse problem takes as inputs a desired position of the effector (piloted by an orange sphere in the figures) and computes the actuation values required to achieve this position (see eq (15)). This allows for the control of the robots in the open loop.

1) Soft finger: Here the inverse problem has a single unknown (the translational force at the base of the cable). But the problem, shown in fig 9, is less trivial than it may seem because all the sliding constraints between the DCM cable model and the FEM model of the silicone finger must also be taken into account.

2) Soft tentacle: With a cable-driven soft tentacle robot made of silicone, we demonstrate the transmission control of several cables, all modeled with DCM. In the experimental scenario shown in Figure 10 we control the tip of the trunk as the effector and the effector follows a predefined trajectory. The robot is actuated with four cables disposed each 90 degrees around the longitudinal direction. In the inverse

(a) pre-bent cable   (b) Finger actuated with pre-bent cable

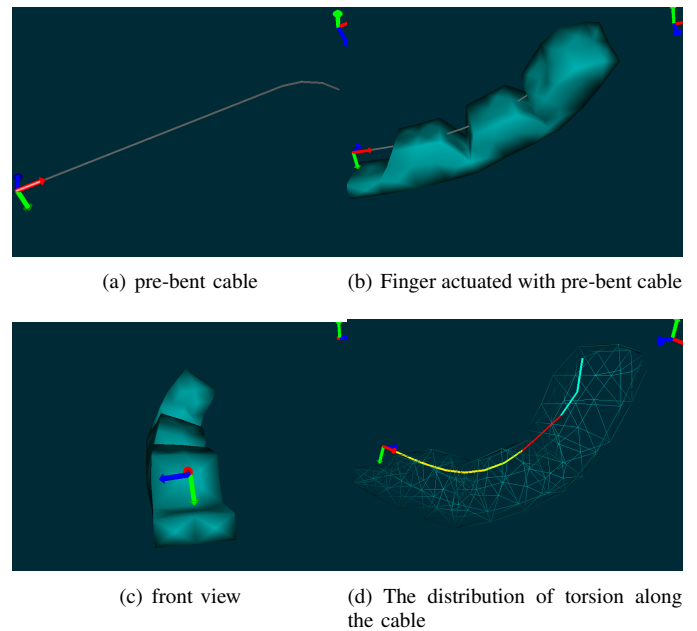(c) front view   (d) The distribution of torsion along the cable

Fig. 8. Use of a pre-bent rod. Like in concentric tube systems, the pre-bent rod deforms the silicone when a rotation is applied in the axis of the rod.
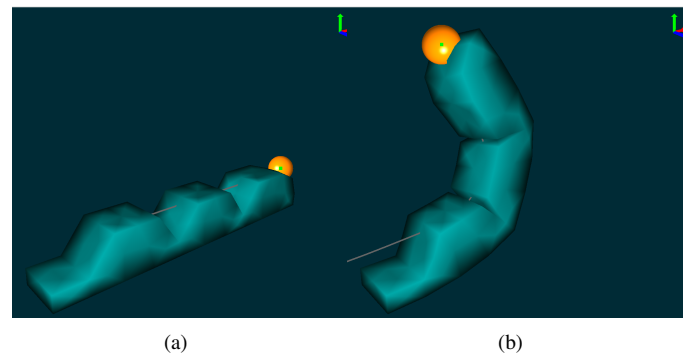

(a)   (b)

Fig. 9. Inverse model simulated on the soft finger model and which allows to find the translation to be applied at the base of the cable to produce the desired displacement on the end effector (orange sphere).

simulation, we demonstrate that we are able to drive the translational motion of the base of the DCM cable model to create the desired motion at the end of the FEM model.

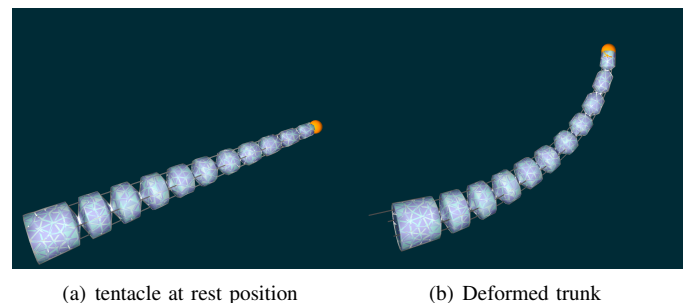(a) tentacle at rest position   (b) Deformed trunk

Fig. 10. Inverse simulation on a soft silicone tentacle modeled in FEM and actuated by 4 cables models in reduced coordinates by DCM.

The computation timing are based on a modern machine (Intel Core i7-8850H CPU @ 2.60GHz × 12).

Simu	#FEM	#DCM	#AC	#SL	#Eff	C	Time(ms)
grasper	1167	45	3	45	0	D	35.97*
Finger	389	15	1	15	3	I	4.31
Trunk	1972	84	4	84	3	I	82.33

TABLE I

SIMU: SIMULATION, **#FEM:** NUMBER OF TETRAHEDRAL MESH, **#DCM** NUMBER OF SECTIONS OF THE DCM MODELE, **#AC:** THE NUMBER OF ACTUATOR CONSTRAINT, **#SL:** THE NUMBER OF SLIDING CONSTRAINT, **#EFF:** THE NUMBER OF EFFECTOR CONSTRAINT, **C:** THE CONTROL (**D**IRECT OR **I**NVERSE), **TIME(MS):** THE COMPUTATION TIME (IN MS) OF ONE SIMULATION STEP. *: THIS TAKES INTO ACCOUNT THE FRICTION AND THE CONTACT RESOLUTION.

It is also important to highlight that the table (table1) allows us to show that we keep a real time computation time as those of [10] where the cable is modeled using a purely geometric constraint without taking into account the constitutive law of the cable material. Using the DCM model (with 8 sections) to control the finger, the computation time is 3.83 ms while the geometric model is 2.54 ms.

V. CONCLUSION AND FUTURE WORK

This work highlights the interest of combining deformable models with global coordinates (such as FEM here) and models with reduced coordinates (such as DCM here) to accurately model the behavior of some soft robots that are operated by cables or rods. In particular, we have shown that it is possible to combine them in a single framework to deduce the direct and inverse models of the robot.

The next steps of this work will be to validate the models obtained on a real robot and in particular to validate the stress levels obtained on the cables. We also wish to use this type of approach to explore new robot designs, using a soft material matrix (such as silicone) and pre-bent rods on which translations and rotations can be applied.

We could also work on improving the speed of computation by combining parallelization, model order reduction for the FEM part [14] and/or recursive algorithms to calculate the DCM.

REFERENCES

[1] D. K. Pai, "Strands: Interactive simulation of thin solids using cosserat models," in *Computer Graphics Forum*, vol. 21, no. 3. Wiley Online Library, 2002, pp. 347–352.

[2] J. Spillmann and M. Teschner, "Corde: Cosserat rod elements for the dynamic simulation of one-dimensional elastic objects," in *Proceedings of the 2007 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, ser. SCA '07, 2007, pp. 63–72. [Online]. Available: http://dl.acm.org/citation.cfm?id=1272690.1272700

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

[4] K. Oliver-Butler, J. Till, and C. Rucker, "Continuum robot stiffness under external loads and prescribed tendon displacements," *IEEE Transactions on Robotics*, vol. 35, no. 2, pp. 403–419, 2019.

[5] D. Trivedi, A. Lotfi, and C. D. Rahn, "Geometrically exact models for soft robotic manipulators," *IEEE Transactions on Robotics*, vol. 24, no. 4, pp. 773–780, 2008.

[6] F. Renda, V. Cacucciolo, J. Dias, and L. Seneviratne, "Discrete cosserat approach for soft robot dynamics: A new piece-wise constant strain model with torsion and shears," in *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2016, pp. 5495–5502.

[7] F. Renda, C. Armanini, V. Lebastard, F. Candelier, and F. Boyer, "A geometric variable-strain approach for static modeling of soft manipulators with tendon and fluidic actuation," *IEEE Robotics and Automation Letters*, vol. 5, no. 3, pp. 4006–4013, 2020.

[8] E. Coevoet, A. Escande, and C. Duriez, "Optimization-based inverse model of soft robots with contact handling," *IEEE Robotics and Automation Letters*, vol. PP, pp. 1–1, 02 2017.

[9] T. Morales Bieze, F. Largilliere, A. Kruszewski, Z. Zhang, R. Merzouki, and C. Duriez, "Finite element method-based kinematics and closed-loop control of soft, continuum manipulators," *soft robotics*, 2018. [Online]. Available: https://hal.archives-ouvertes.fr/hal-01745625

[10] E. Coevoet, T. Morales-Bieze, F. Largilliere, Z. Zhang, M. Thieffry, M. Sanz-Lopez, B. Carrez, D. Marchal, O. Goury, J. Dequidt, and C. Duriez, "Software toolkit for modeling, simulation, and control of soft robots," *Advanced Robotics*, pp. 1–17, 11 2017.

[11] E. Coevoet, A. Escande, and C. Duriez, "Soft robots locomotion and manipulation control using fem simulation and quadratic programming," in *IEEE Robosoft Conference*, 04 2019, pp. 739–745.

[12] M. Nesme, Y. Payan, and F. Faure, "Efficient, physically plausible finite elements," *EUROGRAPHICS*, 2005.

[13] C. A. Felippa, "Iterative procedures for improving penalty function solutions of algebraic systems," *International Journal for Numerical Methods in Engineering*, vol. 12, no. 5, pp. 821–836, 1978. [Online]. Available: http://doi.wiley.com/10.1002/nme.1620120508

[14] O. Goury and C. Duriez, "Fast, generic, and reliable control and simulation of soft robots using model order reduction," *IEEE Transactions on Robotics*, vol. 34, no. 6, pp. 1565–1576, 2018.

[15] T. Morales Bieze, A. Kruszewski, B. Carrez, and C. Duriez, "Design, implementation, and control of a deformable manipulator robot based on a compliant spine," *The International Journal of Robotics Research*, p. 0278364920910487, 2020.

[16] Y. Wang, N. J. Weidner, M. A. Baxter, Y. Hwang, D. M. Kaufman, and S. Sueda, "Redmax: efficient & flexible approach for articulated dynamics," *ACM Transactions on Graphics (TOG)*, vol. 38, no. 4, pp. 1–10, 2019.

[17] Z. Zhang, J. Dequidt, A. Kruszewski, F. Largilliere, and C. Duriez, "Kinematic modeling and observer based control of soft robot using real-time finite element method," in *Intelligent Robots and Systems (IROS), 2016 IEEE/RSJ International Conference on*. IEEE, 2016, pp. 5509–5514.

[18] E. Coevoet, A. Escande, and C. Duriez, "Optimization-based inverse model of soft robots with contact handling," *IEEE Robotics and Automation Letters*, vol. 2, no. 3, pp. 1413–1419, 2017.

[19] F. Faure, C. Duriez, H. Delingette, J. Allard, B. Gilles, S. Marchesseau, H. Talbot, H. Courtecuisse, G. Bousquet, I. Peterlik, and S. Cotin, *SOFA: A Multi-Model Framework for Interactive Physical Simulation*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2012, pp. 283–321. [Online]. Available: http://dx.doi.org/10.1007/8415_2012_125

[20] J. Ferreau, C. Kirches, A. Potschka, H. Bock, and M. Diehl, "qpoases: A parametric active-set algorithm for quadratic programming," *Mathematical Programming Computation*, vol. 6, 12 2014.

[21] T. Hassan, M. Manti, G. Passetti, N. d'Elia, M. Cianchetti, and C. Laschi, "Design and development of a bio-inspired, under-actuated soft gripper," in *Proceeding of the Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*. IEEE, 2015.