Efficient FEM-Based Simulation of Soft Robots Modeled as Kinematic Chains

Maria Pozzi\(^1\), Eder Miguel\(^2,3\), Raphael Deimel\(^4\), Monica Malvezzi\(^1\), Bernd Bickel\(^2\), Oliver Brock\(^5\), Domenico Prattichizzo\(^1\)

Abstract—In the context of robotic manipulation and grasping, the shift from a view that is static (force closure of a single posture) and contact-deprived (only contact for force closure is allowed, everything else is obstacle) towards a view that is dynamic and contact-rich (soft manipulation) has led to an increased interest in soft hands. These hands can easily exploit environmental constraints and object surfaces without risk, and safely interact with humans, but present also some challenges. Designing them is difficult, as well as predicting, modelling, and “programming” their interactions with the objects and the environment. This paper tackles the problem of simulating them in a fast and effective way, leveraging on novel and existing simulation technologies. We present a triple-layered simulation framework where dynamic properties such as stiffness are determined from slow but accurate FEM simulation data once, and then condensed into a lumped parameter model that can be used to fast simulate soft fingers and soft hands. We apply our approach to the simulation of soft pneumatic fingers.

I. INTRODUCTION

The rise of soft and compliant mechanisms \([1], [2]\) leads to a change of understanding of many problems in robotics, because soft robots reimagine the role of hardware, from being precise and accurate (being easy to model) to providing adaptation, robustness, and instant reactions: properties that simplify control. For mechanically complex problems – such as manipulation – the promise of simplified control is especially attractive. But it also means that the morphologies of hands have to become complex. With soft hands, even experts struggle to intuit the ramifications of small design changes, impeding the exploration of the soft hand design space. Simulation could greatly accelerate soft hand design, but only if it is both accurate enough to trust its results and fast enough to enable quick design iteration, i.e. we need to move into the upper left corner in Fig. 1. Today, the most popular simulation approaches are finite element methods (FEM) based on detailed volumetric meshes and multi-body simulation. Both fall short of this goal when applied to soft hands \([3]\).

The main hurdle is to find a trade-off between the computational load and the required accuracy according to the target application. This paper presents a general method for fast and accurate simulation of soft robots that can be approximated by a serial kinematic chain, such as robotic fingers. The method falls into the category of finite-element-based multi-body simulations. We propose an algorithm to extract dynamics parameters for specifying a simplified kinematic chain model of the simulated robot from data gathered with FEM simulations. The obtained values can then be plugged in the lumped parameter model of the robot implemented in standard multi-body simulation tools. With this approach, the accuracy of FEM simulation, which has been often used to simulate flexible mechanisms \([4]–[7]\), is combined with the efficiency and simplicity of a lumped parameter model. In this paper, we investigate the feasibility of exactly such a three-tiered simulation system for PneuFlex actuators \([8]–[10]\), by augmenting the existing, SOFA-based, simulator with a numeric, FEM-based, estimation of the stiffness values required by the simulator. The conversion of the FEM simulation data into discretized stiffnesses for a simplified kinematic model is a central contribution of this paper.

The paper is organized as follows: Sec. II introduces the three-tiered simulation framework. Sec. III then presents in the detail the implementation of each step of the pipeline. In Sec. IV, we apply the entire procedure illustrated in Fig. 2 to the simulation of a PneuFlex actuator and compare the obtained results with experimental data.

II. SIMULATION PIPELINE

Soft robotic hands are based on deformable materials interacting with deformation-limiting structures arranged in a way such that the desired actuation-deformation pairs
are obtained. Predicting the behavior of these combined structures is very difficult due to the typical non-linear behavior of the deformable materials together with the complex interactions with the deformation-limiting structures. This makes the task of designing soft robotic hands difficult and time-consuming and creates the need for tools that provide accurate simulation capabilities for such systems. A recent example of a simulator for hydraulic actuated soft robots can be found in [11].

The starting point of our study is a simulator for soft hands based on soft continuum actuators that achieves interactive simulation rates [10]. It is based on a simplified lumped model that only approximates the actuator deformation, and in which specific parameters of the model cannot be easily defined, due to the complexity of geometries and uncertainties in the definition of material’s mechanical properties. Soft continuum actuators can be more accurately simulated using high-resolution volumetric FEA [6]. Finite Element Method (FEM)-based simulators are the standard tool for realistic and accurate simulation of deformable objects (see [12] for a review on physically-based simulation of deformable objects and [4] for an introduction to FEM-based simulation for deformable objects), but their computational time is still prohibitive.

We believe that the key insight to resolving this dilemma of accuracy and computational feasibility consists in detailing the fast lumped parameter model with more reliable properties, e.g. stiffness values of each node. Such properties could be taken for instance from a stiffness library populated by the results of a suitable set of test cases.

The costly high-resolution FEA can be performed once for any specific geometry and material pair, we then just need to store the results in a database to reuse them whenever the specific actuator and the specific deformation is encountered again, amortising the computational effort spent on the expensive simulation over potentially thousands of simulations.

Fig. 2 shows a general scheme of the proposed simulation environment: the stiffness parameters needed to characterize the kinematic chain model e.g. of each finger of a hand, are extracted from the data gathered with FEA, and then plugged in the lumped parameter models of the hand implemented in grasp simulation tools such as SOFA 1 or SynGrasp 2.

In this paper we focused in particular on the definition of a suitable model reduction methodology for extracting stiffness parameters from FEA results, i.e. a middle layer between FEM models and lumped parameters-based simulators. To define such procedure, we assume that the soft element can be represented as a serial kinematic chain composed of rigid links connected by three dimensional spherical joints. This approximation cannot be applied to soft robots that do not have a principal actuation direction, as for instance the universal gripper [13]. System compliance is concentrated in the joints through the definition of equivalent stiffness values. Since the joints have three degrees of freedom, a

\[ 3 \times 3 \] equivalent stiffness matrix should be defined for each of them; in this paper we assume that such matrices are diagonal, i.e. there are no coupled terms. Elements of such matrices are defined by evaluating from FEA results an estimation of a reduced set of torques and rotations that are assigned to the joints of the corresponding lumped parameter model.

### III. Methodology

This section summarizes the main strengths and limitations of the tools that were used in this paper to simulate pneumatically actuated fingers like those of the RBO Hand 2, an anthropomorphic hand made mainly of silicone rubber [9]. Its five fingers are highly compliant, pneumatic continuum actuators, called PneuFlex [8]; the index, middle, ring, and little finger are 90 mm long and of identical shape, and the thumb actuator is 70 mm long. All fingers get narrower and flatter towards the finger tip and their bottom side contains an inelastic fabric to prohibit elongation. This causes a difference in length between the top and bottom side and, when inflating the contained chamber with air, the pressure forces the hull to elongate along the actuator. Cross-sectional threads surround the finger and stabilize the actuator’s shape (Fig. 3a). The palm consists of two connected actuators and was designed to perform the thumb opposition. Fingers can have different shapes (see [14]), and the ones the hand is composed of are said of type P10.

#### A. FEM Simulation - VegaFEM

We build our FEM-based soft finger simulator on top of VegaFEM 3, a middleware physics library for simulating deformable objects undergoing large deformations. It implements several linear and nonlinear material models and supports model reduction, cloth simulation and rigid body dynamics, it is able to accurately capture the behavior of deformable materials, such as silicone, and provides the base infrastructure to implement additional force models, i.e. actuation forces and deformation-limiting structures.

We focus on the simulation of pneumatically actuated soft fingers consisting on 3 main interacting elements: the

---

1http://www.sofa-framework.org

2http://sirslab.dii.unisi.it/syngrasp

3http://http://run.usc.edu/vega/
base deformable material (based on which the finger body is fabricated), the pneumatic actuation system that drives the deformation, and the passive structures that limit and define the deformation behaviour under actuation. To simulate the base deformable material, we use the NeoHookean material model, where Young’s modulus and Poisson ratio are set to appropriate material parameters (in our examples, \( E = 2.6 \cdot 10^5 \) Pa and \( \nu = 0.48 \)).

For the actuation system, we follow the approach by Skouras et al. [15]. Given a cavity and its internal pressure level, we compute the pressure forces on a FEM node in the enclosing surface as the sum of the pressure force of each incident face:

\[
\mathbf{f}_{\text{pressure}} = \sum_j \frac{1}{3} p_j \mathbf{A}_j \mathbf{n}_j,
\]

where \( f \) traverses all incident faces, \( p \) is the internal pressure and \( \mathbf{A}_j \) and \( \mathbf{n}_j \) are the area and the normal of each incident face, respectively.

Finally, we model the passive structures as stiff springs with zero compression response, which are added to the finger body. Each passive structure is composed of a sequence of links and nodes, where each link represents a stiff spring. In order to compute the forces produced by the passive structures onto the tetrahedral mesh nodes, we first compute the spring forces on the passive structure nodes as:

\[
\mathbf{f}_{\text{spring}} = \sum_l -k \frac{1}{\| \mathbf{v}_l \|} \left( \| \mathbf{v}_l \| - \| \mathbf{v}_0 \| \right) \mathbf{v}_l
\]

where \( l \) traverses all the links incident on the passive structure node, and \( \mathbf{v} \) and \( \mathbf{v}_0 \) are the vectors from any neighbouring node to the passive structure node in the deformed and rest configurations, respectively. Lastly, we transfer the forces to the tetrahedral nodes using the barycentric coordinates of each passive structure node within the tetrahedron containing it. In a typical setup, the passive structures are placed at the bottom side of the finger, to limit the area deformation, and regularly along the main axis, limiting the cross-sectional deformation (see Fig. 3b for an illustration of the passive structures in a deformed finger). This allows us to use isotropic homogeneous materials to represent the body of the finger, closely resembling the real fabricated soft fingers.

The main limitation of our FEM-based simulator is that in order to obtain accurate simulations, high-resolution discretizations are needed, which introduce a significant computational cost and performance slow-down. Additionally, VegaFEM does not implement collision detection or contact handling, hence limiting its applicability to contact scenarios. Using our FEM-based simulator we populate a data exchange layer, where, for a given soft finger design, we store actuation-deformation pairs. Our database generation process starts with a finger design, usually given as a low resolution surface mesh, including the pressure cavity. First, we generate the corresponding high-resolution tetrahedral mesh, attach the passive structures responsible for limiting the deformation and set the deformation model parameters according to the fabrication material parameters. The tetrahedral mesh corresponds to the rest configuration of the simulation mesh and is defined as the concatenation of the \( N \) nodal positions in a column vector, \( \mathbf{X} = (\mathbf{X}_0^T, \mathbf{X}_1^T, \ldots, \mathbf{X}_N^T)^T \).

Then, we generate the actuation samples, \( \mathbf{S} = (S_0, S_1, \ldots, S_M) \), where each actuation \( S_i = (p_i, \mathbf{f}_i^p, \mathbf{f}_i) \) consists on an internal pressure value, \( p_i \), and a set of external forces, defined by the application point \( \mathbf{f}_i^p \), which must lay inside the volumetric mesh, and the force vector, \( \mathbf{f}_i \). For each actuation sample, \( S_i \), we solve a quasi-static problem with boundary conditions defined by the actuation parameters, and obtain the equilibrium configuration of the soft finger as a deformed volumetric mesh, \( \mathbf{x}_i^d = (\mathbf{x}_0^T, \mathbf{x}_1^T, \ldots, \mathbf{x}_N^T)^T \). The pairs \( (S_i, \mathbf{x}_i^d) \) are stored in the database and used later to extract the parameters for the simplified simulation model.

**B. Discrete parameters simulation - SOFA**

SOFA \(^1\) is a software framework to construct multi-model simulations in a modern, modular and object-oriented manner. Simulation state is organized as a scenegraph; which can be constructed and modified at runtime via hooks in a special Python class. For soft hand simulation, a set of Python scripts constructs a discretized, Cosserat-beam based model of a soft hand automatically from a set of actuator parameterizations and a description of the actuator locations. A typical setup discretizes each actuator into ten beam segments, resulting in \( n_{frames} = 10 \cdot 6 \) joints for the whole hand when including one to attach the actuator to the wrist. Thanks to the modular structure of the scene construction code, different actuators can be easily swapped in hands, by exchanging a class encapsulating the computation of mechanical parameters. When the simulator works standalone, it employs a theoretic model published recently [9] to compute the two main properties governing behaviour: joint actuation ratios and joint stiffness matrix.

Due to the low number of DoFs for the complete scene including one hand, one table and one rigid object to grasp (i.e. \( n_{frames} \cdot 6 = 372 \)), the simulation is orders of magnitude faster than a detailed FEM model (ca. \( 0.25 \times \) real time on a single core of an Intel Core i5-6600K CPU @ 3.50GHz [10]), enabling iterative work flows, but also making many consecutive iterations of simulation-driven optimization possible.

---

\(^{1}\) SOFA: Scene-Oriented Freeform Animation
Even though this model gives reasonable results, the computation of both the actuation ratios (6-by-$n_{\text{frames}}$ values) and stiffness matrices (6-by-6-by-$n_{\text{frames}}$ values) relies on many simplifications and implicit assumptions. It is these values that could potentially be supplied by a more elaborate FEM simulation. Unfortunately, appropriate anisotropic material models and caching infrastructure are not available in the SOFA simulator, necessitating the integration with additional simulator codebases.

### C. Stiffness extraction algorithm

The main assumption behind the implementation of the FEM-based multi-body simulation framework we propose in this work, is that we consider soft robots that can be represented as mono-dimensional kinematic chain connecting a set of rigid links with three dimensional spherical joints.

Let us indicate with $J_{i,0}, i = 1, \ldots, n$ the centres of the spherical joints in the reference unloaded configuration (Fig. 4). For each joint $i$, let us indicate with $A_i$ the relative finger cross section area, with $w_i$ its width, $h_i$ its height, and $d_i$ its wall thickness, measured on the cross section passing through $J_{i,0}$ point.

Let $\mathcal{F}_i = \{J_{i,0}, x_0, y_0, z_0\}$ be the first reference frame, its origin is set on the point $J_{i,0}$, corresponding to the center of the first joint. At each joint $i$ we consider a reference frame $\mathcal{F}_{i,0} = \{J_{i,0}, x_i, y_i, z_i\}$ whose origin $J_{i,0}$ is on the joint center and whose axes are oriented as sketched in Fig. 4. The coordinates of the generic $J_{i,0}$ point are indicated with $r_i^0$. For the sake of simplicity we can assume the distance between two adjacent joints as a constant indicated with $a$.

For the finger geometries considered in this paper, the axes of $\mathcal{F}_{i,0}$ frames are parallel.

In the reference configuration, for each joint $J_{i,0}$, we collected all the nodes of the FEM discretization whose positions are included between the planes $x = x_i - b$ and $x = x_i + b$, as sketched in Fig. 4. Let us indicate with $A_i$ the set of FE mesh nodes comprised between these two planes. Different values of $b$ were tested, in particular, its value was set as a function of the distance $a$ between joints. We then evaluate the bounding box $B_i$ containing all these nodes. It is easy to observe that, for the finger geometries considered in this paper, in the undeformed reference configuration, the edges of the bounding box are parallel to the axis of the $\mathcal{F}_{i,0}$ reference frame. In the deformed finger configuration, we indicate with $\psi_i, \theta_i, \phi_i$ the rotations of the $i$-th joint $J_i$ with respect to $x_i$, $y_i$ (secondary bending), and $z_i$ axis (principal bending), respectively. From FEA results we can evaluate how the positions of the FE mesh nodes belonging to subsets $A_i$ displace, and consequently how the bounding box containing them is transformed.

As a first method (Method 1, M1) to evaluate the equivalent rotation angles, we considered that, after the deformation, the edges of the bounding $B_i$ define a new reference frame $\mathcal{F}_i = \{J_i, x_i, y_i, z_i\}$. With respect to the initial configuration, the $i$-th reference frame origin moved from $J_i^0$ to $J_i$ and its axes rotated. Let us indicate with $R_i^0$ the rotation matrix representing such a rotation, with respect to $\mathcal{F}_0$. The local rotation can be evaluated as $R_i = R_{i-1}R_i^0$. Indicating with $r_{ijk}$ its element in position $j, k$, with $j,k = 1, 2, 3$, we can evaluate local equivalent rotation angles $\phi_i, \theta_i, \psi_i$.

Another possible method (Method 2, M2) to extract the mean deformation for each joint from FEM simulation results, consists in representing node displacements through homogeneous matrices. Let us indicate with $p_{j,0}$ the coordinates of the generic node $P_j$ belonging to the section $A_i$ when the finger is in the reference unloaded configuration, and with $p_j$ the coordinates of the same point in a generic deformed configuration, both expressed w.r.t. $\mathcal{F}_0$. We assume that the deformation of the finger can be represented by the following linear relationship

$$\tilde{p}_j = {\bf T}_i p_{j,0}$$

where $\tilde{p}_j$ indicates the homogeneous representation of $p_j$, i.e. $\tilde{p}_j = [p_j^T \, 1]^T$. We furthermore assume that $\bf T_i$ can be approximately considered constant over each section $A_i$. In this matrix, it is straightforwardly possible to extract the vector $u_i$, representing the translational part, and the $3 \times 3$ matrix $A_i$, that includes both the rigid rotation and the deformation. From FEA results, $p_j$ and $p_{j,0}$ coordinates are given for each node of each section, so from eq. (1) a suitably defined linear system can be defined for each section whose solution provides an estimation of matrix $\bf T_i$. We furthermore elaborated, through the polar decomposition theorem, as the product of an orthogonal matrix $R_i$ and a symmetric and positive definite matrix $U_i$, i.e. $A_i = R_i U_i$, with $U_i = \sqrt{A_i^T A_i}$ and $R_i = A_i U_i^{-1}$. The analysis of $U_i$ terms could give interesting results in terms of finger deformation, however they are beyond the scope of this paper. From the rotation matrix $R_i$ we can easily evaluate the equivalent rotation angles $\phi_i, \theta_i, \psi_i$, as previously introduced. In the generic loading condition, let us indicate with $p$ the inflating pressure, and with $f \in \mathbb{R}^{3n}$, the vector containing the external forces $f_i, k = 1, \ldots, n$, applied to the finger in the joints, whose components are expressed w.r.t. $\mathcal{F}_0$. The corresponding equivalent joint torques can be expressed as

$$\tau = J^T f + J p \rho$$

where $\tau = [\tau_1, \ldots, \tau_n]^T$, and $\rho = \tau_{x,i}, \tau_{y,i}, \tau_{z,i})^T$, $J$ is a $3n \times 3n$ matrix representing the finger Jacobian matrix evaluated with respect to the force application points $F_e$, and $J_{ij} \in \mathbb{R}^{3n}$ is a vector that allows to evaluate joint torque actions equivalent to the application of a given inflating pressure, its elements depend on cross sectional geometric properties, that determine the effective air chamber area.

In each joint we introduced a three dimensional stiffness element in which we concentrated the compliance of finger elements. Let us indicate with $k_{\psi,i}, k_{\theta,i}, k_{\phi,i}$ the angular stiffness value for each joint of the discretization. Such values relate the joint relative rotation to the equivalent torque, i.e. $\tau_i = K_i \phi_i$, where $\phi_i = [\psi_i, \theta_i, \phi_i]^T$, and $K_i = \text{diag} ([k_{\psi,i}, k_{\theta,i}, k_{\phi,i}])$. For the sake of simplicity, we assumed that each local stiffness matrix $K_i$ is diagonal.
Collecting all the relative rotations in a vector $\Phi$, and the stiffness matrices in the matrix $K = \text{diag}([K_1, \ldots, K_n]) = \text{diag}([k_1, \ldots, k_{3n}])$, we can express the joint torques as $\tau = K\Phi$. Then, since $K$ is diagonal, we can decouple such equations as $\tau_j = k_j\phi_j$ and obtain the elements of $K$ as $k_j = \tau_j/\phi_j$, with $j = 1, \ldots, 3n$.

IV. SIMULATION OF A FINGER

In this section we apply the two stiffness extraction algorithms to three different finger geometries: i) a PneuFlex actuator of type P10 (see Fig. 3a and Fig. 6d), with wall thickness $d = 2.5$ mm ii) a parallelepiped with constant cross section along its length and same wall thickness and material properties of P10, and iii) the finger type P13, with a geometry similar to P10, but longer and with thicker walls ($d = 6$ mm). For all the cases, the material was modelled with Young’s Modulus $E = 2.6 \times 10^5$ Pa and Poisson’s ratio $\nu = 0.48$. For the P10 design, we also compare the simulation pipeline results with ground truth measurements.

A. Results of the stiffness extraction

We analysed three main movements of the soft finger subject to external loading: principal bending (flexion in the actuated direction), lateral bending, and torsion. These motions correspond respectively to three rotations in the considered reference frame (Fig. 4): around the $z$-axis, the $y$-axis, and around the $x$-axis, respectively. To evaluate the stiffness extraction algorithm proposed in this paper, we computed the stiffness values with both methods from Sec. III-C, and for each of them we used three different resolutions $b$ for the computation of the bounding boxes. With each method, we tested three bounding box sizes, with $a = 1$ cm and $b = a, \frac{a}{2}, \frac{a}{4}$. The models therefore had 9, 9 and 10 joints for P10, parallelepiped, and P13, respectively.

Result: Stiffness in the principal bending direction, as expected, is constant for regular shapes, and higher for fingers with thicker walls. To evaluate the stiffness of the finger along the $z$-axis, $k_\phi$, we analysed the FEM simulations considering as actuation only the inflating pressure, varying from 10 kPa to 80 kPa. Results for the principal bending of the P10 with the two methods are reported in Fig. 5a and Fig. 5b. They can be compared with those obtained for a parallelepiped finger (Figs. 6a, 6b), and for the P13 finger (Figs. 6c). In P10 and P13, the stiffness values do not change significantly between the two methods and decrease with the inflated pressure. $k_\phi$ is higher for the initial nodes and lower for the final nodes due to the fact that one edge of the finger is fixed during FEM simulations and that the initial part of the finger is filled. The stiffness tends to decrease along the finger due to the height reduction, that reduces the area moment of inertia of the cross section. It is worth noticing that the stiffness of P13 is almost twice the one of the P10, due to its different wall thickness. The parallelepiped finger presents a similar behaviour to the other two in the initial joint, but, due to its regular shape, its stiffness tends to remain approximately constant across the other joints. This is visible in Method 2, and less visible in Method 1, which is more sensitive to discretization.

Result: Method 1 is more sensitive to discretization parameters. To observe the sensitivity of $k_\phi$ to the resolution $b$, we computed it with the two different methods for a fixed value of internal pressure (80 kPa) and for different values of $b$. Results are shown in Fig. 5c. When the stiffness is computed with Method 1 (red) and with $b = a$, it is slightly higher with respect to the other cases, but the other curves have very similar values.

Result: Lateral stiffness computation can encounter problems due to buckling and coupled torsional motions. To compute the stiffness in the lateral direction (Fig. 7), $k_\psi$, only FEM samples with $p = 0$ Pa and $f_{ext} = [0, 0, f_z]^T$ with $f_z \neq 0$ N were used. The stiffness profiles obtained in this case have a behaviour similar to the principal bending ones, the stiffness decreases as the finger height decreases. For the lateral bending cases, when the force applied to the fingertip is higher than 0.7 N, in FEM simulations we observed a macroscopic buckling of the finger, in this case the model reduction application leads to large errors (Fig. 7a). The lateral stiffness values computed with Method 2 (Fig. 7b) presented problems when small applied forces were considered; in this case the rotation is quite small and the main bending contribution is coupled to torsional motions. $k_\psi$ here is higher than in Fig. 7a. This could be a reason why M2 overestimates the torque in the $y$-axis in the final experiments (see 11b).

Result: Torsional stiffness computation: similar trend with both methods. The stiffness in the torsional axis, $k_\psi$, was computed considering only FEM samples with $p = 0$ Pa and $f_{ext} = [0, f_y, 0]^T$ with $f_y \neq 0$ N were considered. $k_\psi$ was found to be higher in the first nodes, the closest to the point where the finger is fixed, and lower in the last nodes. The trend of torsional stiffness values doesn’t vary significantly between the two methods, so we just reported what we obtained with M1 (Fig. 5d).

B. Validation of the simulation tools against real data

When validating the stiffness parameters of a simulation against real measurements, we face a problem: we need...
the actuator’s curvature to compute stiffnesses and actuation ratios. Unfortunately, the extreme sensitivity of curvature computation to marker position prevents us from doing meaningful comparisons of local curvatures between model and experimental data. To circumvent this problem, we operate on aggregates of segment motion such as the fingertip position and orientation instead. This is a straightforward approach for the actuated axis, as we can avoid rotation around the other two orthogonal axes. It has, e.g., been used to validate the analytic model [9], [10], and in this paper to validate the principal bending simulated in FEM.

**Result:** VegaFEM simulation well predicts the principal bending. Results in Fig. 8a show that the FEM simulation done with VegaFEM is able to capture the behaviour of the real experiments for the principal bending of the P10 finger, because the orientation predicted with FEM is very close to the experimental data recorded in [9]. Since the final material parameters of the silicone used to fabricate the finger body can vary significantly due to the fabrication process, we compare the experimental data with a set of simulations using Young’s Modulus $E$ in a realistic range for the *Dragonskin* silicone used in our experiments: $[2.5e5, 3.0e5]$ Pa.

For $x$– and $y$– axes the deformation will not be restricted to a 2D plane as the finger has to be bent around the actuation axis too. Therefore we need to analyse the full 3-dimensional deformation. To this aim we set up a P10 PneuFlex finger in a motion capture system (Motion Analysis Osprey, at $100\,Hz$), including a Force/Torque sensor attached to the base of the finger. The finger was inflated to bend the fingertip by $90^\circ$, and then forces were applied to the fingertip via a point contact facilitated by a chopstick. Both the fingertip motion and fingertip base were tracked using L-frames. Fig. 9a shows the experimental setup. For the evaluation, the fingertip was moved left and right ($0s$ to $10s$), down and up ($25s$ to $45s$) two times each. Finally
In both cases we use stiffness in Figs. 10 and 11, for the stiffness computed with Method 1 and Method 2, respectively. In both cases we use stiffness in Figs. 10 and 11, for the stiffness computed with Method 1 and Method 2, respectively. In both cases we use stiffness in Figs. 10 and 11, for the stiffness computed with Method 1 and Method 2, respectively. In both cases we use stiffness in Figs. 10 and 11, for the stiffness computed with Method 1 and Method 2, respectively. In both cases we use stiffness in Figs. 10 and 11, for the stiffness computed with Method 1 and Method 2, respectively. In both cases we use stiffness in Figs. 10 and 11, for the stiffness computed with Method 1 and Method 2, respectively.

Result: For the torque $T_x$ both the baseline and our FEM-based model are almost identical.

Result: For $T_y$ both our FEM-based and the baseline model show considerable deviations (too soft). This is due to the fact that the movement in the lateral direction is always coupled with a torsion and a bending, making it difficult to compute the stiffness in the $y$ direction.

Result: For $T_z$, our FEM-based model fits the ground truth well, above all with Method 1, while the baseline greatly overestimates the finger stiffness.

V. Conclusions and Future Work

In this work we present a pipeline for fast and accurate simulation of soft robots that can be modelled with serial kinematic chains. Slow and accurate FEM simulations are generated once and then processed to obtain an equivalent stiffness matrix, that can be plugged in the lumped parameter model in a multi-body simulator like SOFA. We applied this framework to a soft pneumatic actuator and we showed that by analysing less than 30 FEM samples, namely those with null applied force for computing the $z$-axis stiffness, and those with null inflated pressure for the $y$ and $x$-axes stiffnesses, we obtained a good accuracy in predicting real finger behaviour. This came at very low computational cost, considering that it takes about 1.5 hours for simulating with FEM a finger discretized with $\approx 20K$ elements, and about 6 ms for simulating a single finger in SOFA. Notwithstanding its advantages, our method presents some limitations, including that it is not able to predict non linear movements such as buckling, or coupled motions. Additional accuracy could be gained by analysing more FEM samples to cover a significant portion of the parameter space or by extracting also other dynamics parameters, including inertia properties and damping. In this case, the three–layered approach can still be adopted, by choosing reduction techniques that take into account not only structure static deformation, but also its dynamic behaviour [16]. As a future development, we plan to test and compare other simulation approaches. For instance, an interesting solution is proposed in [17], where the robot is divided into sub-parts modelled using FEM, and accurate FEA results are used to evaluate equivalent stiffness matrices.

This paper is a first step towards efficient simulations of soft mechanisms by incrementally abstracting and approximating detailed phenomena to focus on those aspects of the behaviour that are relevant to the task. Future applications range from interactive design, allowing rapid visual inspection of the consequences of a design decision, to simulation guided optimization and dynamic simulation of soft hands.

REFERENCES

Fig. 10: (a) Frame of the F/T sensor in the finger base. (b)-(g) Torques $T_x, T_y, T_z$ predicted by baseline simulation (---), predicted by FEM-based simulation with M1 (coloured), and measured by the F/T sensor (black), are plotted with respect to time for the entire test and in the key areas: lateral bending (0–10 s), principal bending (25–45 s), and torsion (50–70 s).

Fig. 11: Torques computed with FEM-based simulation with M2 (coloured), and F/T sensor (black).