Calibration method for soft robots modeled with FEM: application to anisotropy

Félix Vanneste, Olivier Goury, Christian Duriez

Calibration method for soft robots modeled with FEM: application to anisotropy

Félix Vanneste*, Olivier Goury*, Christian Duriez*

***Abstract*—This paper aims at contributing to the sim2real challenge in soft robotics. We present a method for automatic finite element model calibration based on real data using quadratic programming optimisation. The method is generic and evaluated in this study to fit mechanical parameters from anisotropic materials. We show that we are able to optimise the mechanical properties of a given structure along its shape to achieve a given configuration goal. We show the methods interest for calibration by taking reference points from a real world robot and use them in our optimisation process as goals the simulation has to match. Our process will minimise the errors introduced by manufacturing, imperfect models or even mechanical fatigue/plasticity.**

***Index Terms*—Calibration and Identification, Optimization and Optimal Control, Modeling, Control, and Learning for Soft Robots, Soft Robot Materials and Design**

I. INTRODUCTION

SIM2REAL is one of the key challenge in robotics, and perhaps even more so in soft robotics. By having an accurate physical model of a robot, robust control methods can be derived, but also the robot can be simulated in its operating environment to plan and learn automatically [1][2]. In soft robotics, the challenge of having an accurate physical model is even more important because it must take into account deformations, which are derived from sometimes complex material properties. Being able to precisely calibrate these models by optimising the material parameters is certainly a step forward towards better sim2real transitions in soft robotics.

In the community, the question of using differentiable models has had recent interest since it allows gradient-based optimisation to improve accuracy, in particular to fit real data. This was especially investigated in machine learning approaches which can be easily differentiated, for example [3, 4]. In [5], a differentiable model based on material point method was proposed. In a finite element modeling (FEM) context, an analytical gradient with respect to design variables is used in [6] to optimise sensors positioning. In the particular context of soft robotics materials, [7] proposes to fit a model with data from several mechanical tensile tests, thanks to an analytical gradient and a quasi-Newton optimisation method. It was successfully applied for hyperelastic models. Similarly in [8], FEM simulations are used to fit data obtained with simple tensile tests. See also [9] for a more general review on differentiable simulation.

In this study, we focus on differentiable models for soft robots made of anisotropic materials. The main difference with previously mentioned work is that it does not require to bother with fitting an FEM of the tensile test that was performed, which can be cumbersome. Instead, we rely on the presence of sensors to directly fit the material parameters used in the FEM of the robot. We also consider the case of anisotropy, which implies differentiating with respect to anisotropy direction. Finally, we optimise with respect to several robot configurations and provide insights on the conditioning of the optimisation problem. The anisotropic structure of the robot is created by using a 3D printed meso-structure [10].

In our previous work, we have shown that we can use FEM and homogenisation to reproduce the behavior of the structure [11]. But homogenisation is an approximation and we pointed out that the printing method (by extrusion) could add additional anisotropies along the printing direction. Once the robot structure is 3D-printed, it is therefore useful to be able to optimise the parameters of the anisotropy model from data that can be collected directly on the robot. The objective is to ensure that the mechanical model fits the data as closely as possible.

This paper proposes an efficient convex optimisation method to determine the parameters of a deformable FEM, based on the behavior of this model in several configurations of its workspace. Although this work is carried out in the context of manufacturing soft robots from anisotropic mesostructures, we would like to stress that the approach presented below is generic and could be easily extended to calibrate other deformable models based on other constitutive laws.

The contributions of the paper are the following:

- Formulation of the material parameters optimisation problem as a convex optimisation over several configurations of a soft robot.
- Application of the method on anisotropic mechanical parameters.
- Demonstration of the method in the use case of simulation calibration for more accurate open-loop control of an anisotropic soft robot.

Manuscript received: October, 11, 2021; Revised January, 12, 2022; Accepted February, 9, 2022.

This paper was recommended for publication by Editor Cecilia Laschi upon evaluation of the Associate Editor and Reviewers' comments.

* University of Lille, Inria, CNRS, Centrale Lille, UMR 9189 CRIStAL, F-59000 Lille, France (DEFROST team)

Digital Object Identifier (DOI): see top of this page.

II. Background : optimisation-based inverse model

In this section, we present the FEM approach used to model anisotropic soft robots and the convex optimisation used on this model to perform open loop control.

A. Static formulation with constraints

In this paper we consider the static equilibrium of the soft robot in its current configuration. This configuration is represented by the position vector $\mathbf{x}$ of the nodes of the FEM. On any equilibrium configuration, we have:

$$\mathbf{f}_{\text{ext}} - \mathbf{f}(\mathbf{x}) + \mathbf{H}_a^T \boldsymbol{\lambda}_a = \mathbf{0} \tag{1}$$

In this equation, $\mathbf{f}(\mathbf{x})$ is the vector of the non-linear internal forces of the deformable structure, computed by the FEM, and $\mathbf{f}_{\text{ext}}$ is the vector of the external forces f_{ext}, such as gravity forces. We use a Lagrangian formulation to model the loads exerted by the actuators $\mathbf{H}_a^T \boldsymbol{\lambda}_a$, with $\mathbf{H}_a^T$ the Jacobian matrix of the actuation and $\boldsymbol{\lambda}_a$ the vector of the Lagrange multipliers representing the forces exerted by the different actuators.

When the static equilibrium is perturbed (by a change in the loads exerted by the actuators), we will classically try to find a new equilibrium position. To perform this non-linear computation, a limited series expansion of the forces around the current configuration allows to bring back the problem to a (changing) linear formulation:

$$\mathbf{f}(\mathbf{x} + d\mathbf{x}) \approx \mathbf{f}(\mathbf{x}) + \underbrace{\frac{\partial \mathbf{f}}{\partial \mathbf{x}}}_{\mathbf{K}(\mathbf{x})} d\mathbf{x} \tag{2}$$

Where $\mathbf{K}(\mathbf{x})$ is the tangent stiffness matrix, that depends on the current position of the FEM nodes. In an iterative approach to find the static equilibrium, $d\mathbf{x}$ corresponds to the displacement between two successive positions $d\mathbf{x} = \mathbf{x}_i - \mathbf{x}_{i-1}$. So at each iteration, we find $d\mathbf{x}$ and $\boldsymbol{\lambda}_a$ by solving this system:

$$\mathbf{K}(\mathbf{x}_{i-1}) d\mathbf{x} = \mathbf{f}_{\text{ext}} - \mathbf{f}(\mathbf{x}_{i-1}) + \mathbf{H}_a^T \boldsymbol{\lambda}_a \tag{3}$$

Depending on the case, the way of obtaining $\boldsymbol{\lambda}_a$ varies. If we impose the actuator forces $\boldsymbol{\lambda}_a$, the problem is straightforward. If the actuators positions are imposed, additional equations corresponding to the actuators motion are set and $\boldsymbol{\lambda}_a$ plays the role of Lagrange multipliers for these new equations. Note that to compute the FEM, we use the open source SOFA[1] framework.

B. Optimisation for open loop control

In the case of inverse problem (we now look for the actuation loads to match specific goals, rather than applying known value of $\boldsymbol{\lambda}_a$ like in the direct problem), the two unknowns of Eq. (3) remain the same. But additionally, a new set of equations is defined to control the shape of the robot through specific points on the FEM mesh, called effectors.

[1]https://www.sofa-framework.org/

The distance of these effectors to their goal positions (that corresponds to the input of the optimisation) are measured by $\boldsymbol{\delta}_e$. Over a simulation step, this distance can be defined by a linear approximation of the nodes displacement, using FEM interpolation:

$$\boldsymbol{\delta}_e = \mathbf{H}_e d\mathbf{x} + \boldsymbol{\delta}_e^0 \tag{4}$$

with $\mathbf{H}_e$ being a (highly sparse) rectangular matrix, and $\boldsymbol{\delta}_e^0$ the distance vector, evaluated at the beginning of the step. Combining equations (3) and (4), we can derive the relationship between $\boldsymbol{\delta}_e$ and $\boldsymbol{\lambda}_a$ by:

$$\boldsymbol{\delta}_e = \underbrace{[\mathbf{H}_e \mathbf{K}^{-1} \mathbf{H}_a^T]}_{\mathbf{W}_{ea}} \boldsymbol{\lambda}_a + \underbrace{\mathbf{H}_e \mathbf{K}^{-1} (\mathbf{f}_{\text{ext}} - \mathbf{f}(\mathbf{x}_{i-1})) + \boldsymbol{\delta}_e^0}_{\boldsymbol{\delta}_e^{\text{free}}} \tag{5}$$

We now use this relation in a quadratic programming (QP) [12] optimisation scheme in order to find the $\boldsymbol{\lambda}_a$ which allows to minimise $\boldsymbol{\delta}_e$, ie: to find how to actuate the robot so that selected points reach the different goal positions:

$$\begin{cases} \min \|\mathbf{W}_{ea} \boldsymbol{\lambda}_a + \boldsymbol{\delta}_e^{\text{free}}\|^2 \\ \text{with } \boldsymbol{\delta}_{min} \leq \boldsymbol{\delta}_a = \mathbf{W}_{aa} \boldsymbol{\lambda}_a + \boldsymbol{\delta}_a^{\text{free}} \leq \boldsymbol{\delta}_{max} \\ (\text{and } \boldsymbol{\lambda}_a \geq 0) \end{cases} \tag{6}$$

with $\boldsymbol{\delta}_a$ being the actual displacement of the actuator and $\boldsymbol{\delta}_{min}$/$\boldsymbol{\delta}_{max}$ its limits. W_{aa} is homogeneous to a compliance matrix, it gives the influence of the different actuators on themselves, it allows to see how they are coupled/decoupled. $\boldsymbol{\lambda}_a \geq 0$ is used in case of unilateral actuation (for instance cable actuators). To have a well posed convex problem which converge to a unique solution and avoid instabilities, the QP matrix $\mathbf{W}_{ea}^T \mathbf{W}_{ea}$ has to be positive-definite. To have this property, it is necessary (but not sufficient) that the size of the actuator space is equal or less than the size of the effector space.

III. Differentiation and optimisation on multiple configurations

In the model calibration problem, we aim at estimating the mechanical parameters that best fit the robot that was actually built. To verify that the model fits the data, we can continue to use an optimisation based on a set of effector points placed on the robot. However, this time, it is not about optimising the force on the actuators but the parameters of the FEM. In this section, we present the differentiation of the internal forces by these parameters, to make new Lagrange multipliers appear. Then, we show how to adapt the optimisation on several configurations, in order to calibrate the model on several points of the robot workspace.

A. Structural characterisation parameters

The mechanical parameters $\mathbf{p}$ of the constitutive law influence the computation of the internal forces. To use our method, these parameters need to be differentiable, which is the case for the anisotropic case described below.

$$\mathbf{f}(\mathbf{x}+d\mathbf{x},\mathbf{p}+d\mathbf{p}) \approx \mathbf{f}(\mathbf{x},\mathbf{p}) + \underbrace{\frac{\partial \mathbf{f}}{\partial \mathbf{x}}}_{\mathbf{K}(\mathbf{x},\mathbf{p})} d\mathbf{x} + \frac{\partial \mathbf{f}}{\partial \mathbf{p}} d\mathbf{p} \quad (7)$$

In practice, we perform a numerical differentiation of the internal forces to obtain the value of $\mathbf{H}_p^T$ which is equal to $\frac{\partial \mathbf{f}}{\partial \mathbf{p}}$. Written like this, the variation of the parameters $d\mathbf{p}$ can be assimilated to Lagrange multipliers $\boldsymbol{\lambda}_p$. Additionally, to have parameters working in a uniform space (for example between -1 and 1), we can apply scaling factors. For consistency of notations, we introduce $\mathbf{S}\boldsymbol{\lambda}_p = d\mathbf{p}$, where $\mathbf{S}$ is a diagonal scaling matrix allowing to have all parameters in a uniform space and $\mathbf{H}_p^T(\mathbf{x},\mathbf{p}) = \frac{\partial \mathbf{f}}{\partial \mathbf{p}}\mathbf{S}$.

Combining with Eq. (3) we now have the quasi-static equilibrium defined by:

$$\mathbf{K}(\mathbf{x}_{i-1},\mathbf{p}_{i-1})d\mathbf{x} = \mathbf{f}_{\text{ext}} - \mathbf{f}(\mathbf{x}_{i-1},\mathbf{p}_{i-1}) + \mathbf{H}_a^T\boldsymbol{\lambda}_a + \mathbf{H}_p^T\boldsymbol{\lambda}_p \quad (8)$$

Once the optimisation problem (which will be presented below) is solved, we update the parameter $\mathbf{p}_i$ using $\mathbf{p}_i = \mathbf{p}_{i-1} + \mathbf{S}\boldsymbol{\lambda}_p$ and determine the new $d\mathbf{x}$ then continue to iterate to progressively reach an equilibrium ($d\mathbf{x} = 0$ and $\boldsymbol{\lambda}_p = 0$). Note that in the case of a robot calibration, we will impose references to the actuators (so we know $\boldsymbol{\lambda}_a$ or $\boldsymbol{\delta}_a$).

B. Multi-Configuration optimisation

The goal of the optimisation is to update the parameters $\mathbf{p}$ so that the positions of the effector points on equilibrium are as close to the data as possible. But, unlike the optimisation presented in Eq. (6), we seek to optimise the parameters over several different static equilibrium configurations, representative of the robot workspace.

For each of these configurations j, we will launch a different simulation by fixing the material parameters at a starting value but choosing different value of actuation $\boldsymbol{\lambda}_a^j$ (or $\boldsymbol{\delta}_a^j$). Each simulation compute the FEM separately (each of them solves a different version of Eq. (3)). When the equilibrium is reached for each configuration, we start the optimisation of the parameters which will link them all together. Indeed the modification of the parameters has repercussions on the static equilibrium for all the configurations. To conduct the optimisation, we will look at the errors on the effectors at the level of each j configuration, and we will do the same type of condensation as the one done in Eq. (5):

$$\boldsymbol{\delta}_e^j = \mathbf{W}_{ep}^j\boldsymbol{\lambda}_p + \mathbf{W}_{ea}^j\boldsymbol{\lambda}_a^j + \boldsymbol{\delta}_e^{\text{free},j} \quad (9)$$

Note that $\boldsymbol{\lambda}_p$ is the only term that do not depend on the configuration since we look for one set of material parameters that is identical across all configurations, and again that $\boldsymbol{\lambda}_a^j$ is known (or we can define equality equations to impose $\boldsymbol{\delta}_a^j$ by computing $\boldsymbol{\lambda}_a^j$).

Let's suppose that we perform the optimisation on n configurations, we can aggregate these equations:

$$\underbrace{\begin{bmatrix}\boldsymbol{\delta}_e^1\\ \vdots \\ \boldsymbol{\delta}_e^n\end{bmatrix}}_{\boldsymbol{\Delta}_e} = \underbrace{\begin{bmatrix}\mathbf{W}_{ep}^1\\ \vdots \\ \mathbf{W}_{ep}^n\end{bmatrix}}_{\mathbb{W}_{ep}}\boldsymbol{\lambda}_p + \underbrace{\begin{bmatrix}\mathbf{W}_{ea}^1\boldsymbol{\lambda}_a^1 + \boldsymbol{\delta}_e^{\text{free},1}\\ \vdots \\ \mathbf{W}_{ea}^n\boldsymbol{\lambda}_a^n + \boldsymbol{\delta}_e^{\text{free},n}\end{bmatrix}}_{\boldsymbol{\Delta}_e^0} \quad (10)$$

The vector $\boldsymbol{\Delta}_e$ gathers the effectors distances to the objectives on all configurations. The goal of the minimisation is to find the set of parameters $\boldsymbol{\lambda}_p$ which minimise the square norm of this vector:

$$\min \|\mathbb{W}_{ep}\boldsymbol{\lambda}_p + \boldsymbol{\Delta}_e^0\|^2 \quad (11)$$

When a new value of $\boldsymbol{\lambda}_p$ is found, the parameters $\mathbf{p}$ are updated and each configuration can update its quasi-static equilibrium. This ends a simulation step. It is repeated a number of steps until $\boldsymbol{\lambda}_p \to \mathbf{0}$, we reach a minima and stop the optimisation.

IV. Parameterization of anisotropy

In this section we present how we define varying mechanical properties of a mesh to create transverse behavior with as little parameters as possible in order to simplify later on the optimisation presented in III.B. Then we explain how we pass from these parametric mechanical properties of our mesh into a real 3D printed part.

A. Simulation - Mechanical structure

We optimise the mechanical parameters over a FEM. In this paper we focus on a special case of anisotropy which is transversely isotropic material. We have 5 mechanical parameters to define the elasticity tensor of any element of the mesh : 2 Young modulus ($\boldsymbol{E_t}, \boldsymbol{E_l}$) , 2 Poisson ratio ($\boldsymbol{\nu_t}, \boldsymbol{\nu_{tp}}$) and one shear modulus ($\boldsymbol{\mu_l}$).

Using the approach of [13], we can reduce the parameters needed to only $\boldsymbol{E_t}, \boldsymbol{E_l}$ and $\boldsymbol{\nu_t}$ while insuring a stable anisotropy. To simplify further, we focus on optimising only the Young modulus by setting $\boldsymbol{\nu_t}$ to a chosen value.

In addition to these mechanical parameters we also add a last parameter which is the "fiber" direction defined by a unit vector called $\vec{\boldsymbol{u}}$. With the printing technique used (presented in IV.B), we can only configure the fiber in the printing plan. As a consequence we use one rotation parameter to define $\vec{u}$, we will call it $\boldsymbol{\theta}$. In the end, we can define a transversely isotropic tetrahedron with only 3 parameters: $\{\boldsymbol{E_t}, \boldsymbol{E_l}, \boldsymbol{\theta}\}$.

Even with these simplifications, we cannot optimise these 3 parameters for each tetrahedron of the mesh: the number of unknowns would be too high and dependent on the mesh resolution. Moreover there is a necessary continuity of parameters between neighboring elements. This is why we place what we call control points (**CP**) upon the mesh, we can see on Fig. 1 a visual representation of them (red dots).

Each *CP* will have the triplet of parameters presented above and they will be independent from each other. Then the parameters of each *CP* are diffused on the mesh: for each tetrahedron the triplet of mechanical parameters are obtained

by interpolation, depending on their relative proximity to the different *CP*. For that we use a simple spacial interpolation called Inverse Distance Weighting (**IDW**) [14] that allows to have a smooth transition between the different values while never going above the values it interpolate on. Using the *CP* and doing an *IDW* between them we only have to optimise 3 parameters times the number of *CP* used.

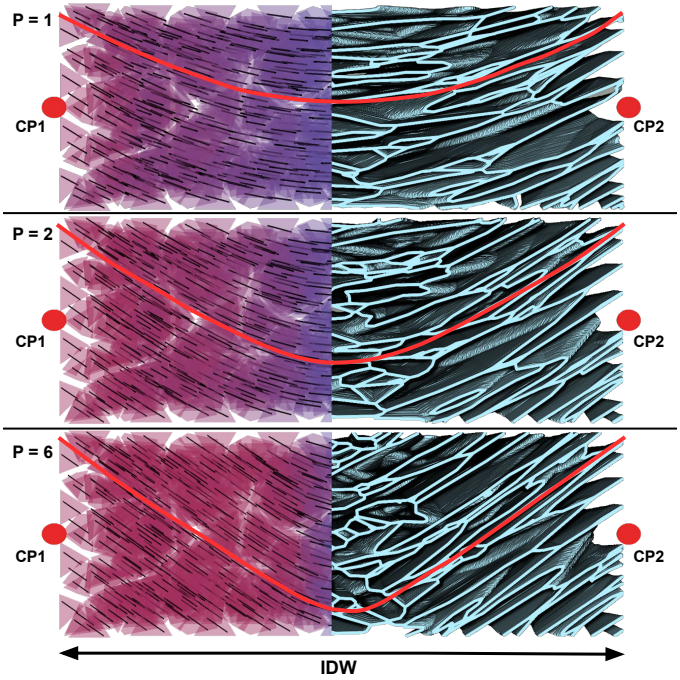

Fig. 1: Interpolation between 2 *CP* (red dot) with, on the left each little black lines corresponding to a tetrahedron's direction and on the right we have the corresponding 3D shape generated. The right/left CP have respectively a set direction of $-35°/35°$. There is also a comparison with different values of p in the IDW interpolation [14].

B. Fabrication - Slicing configuration

As presented in previous work [11], to build the soft structure of the robot we use classical fused filament fabrication (FFF) with a thermoplastic polyurethane filament (TPU) called NinjaFlex (Shore Hardness: 85A and a tensile modulus of 12MPa : data from the constructor) with 1.75 mm of diameter.

To be able to generate a nearly transversely isotropic material and control its different mechanical parameters we use the slicer IceSL[2]. This slicer has many interesting options including the ability to "paint" on a given shape a desired flexibility. We can specify in 3D the coefficient of anisotropy as well as a direction and a density.

We can see on Fig. 1 different interpolation results between two *CP* for different values of the power parameter p used in the *IDW* (see [14]). It shows that with $p = 1$ the transition is too smooth and with $p = 6$ too sharp. As a middle ground we choose to use for all the presented examples in this paper a value of 2. This figure also demonstrate that we have a close match between the different directions put into the simulation and reality.

[2]https://icesl.loria.fr

V. Problem Conditioning

The approach proposed in this paper is based on a convex optimisation. This type of optimisation is sensitive to conditioning, which can be seen as a drawback. On the contrary, we show in this section that an in-depth analysis of the matrices conditioning allow to get practical and intuitive information, useful for the calibration. Given the number and the type of parameters, are there enough effector points on the structure? Are there enough configurations tested in the workspace? Were all parameters excited in the configurations so they can be identified?

As presented in IV.A, the number of parameters is directly proportional to the number of *CP*. If the number of *CP* is high, the number of unknown parameters is high and the data collected during the calibration has to be rich enough to insure convergence during the optimisation. On the other hand, having few *CP* is not enough to have a well posed problem: indeed if we place two *CP* too close to each other, especially on a zone which deforms little, then the set of parameters of these two *CP* will be redundant and the problem badly posed.

The input data can vary in two ways: we can place several effector points on the robot and we can choose several configurations on which we do the calibration. All these choices will have a direct impact on the convergence of the parameters, we will study this in the following.

A. Sample Problem setup

To illustrate the mathematical analysis, we will use the relatively simple example of optimising the anisotropy parameters on a cantilever soft anisotropic beam. This beam has a rectangular section and we place on its 2 extremities a *CP*. We suppose that the mass is known and we optimise only the two elasticity parameters and the anisotropy direction so $\mathbf{p}$ to be optimised is a vector of dimension 6 ($dim(\mathbf{p}) = 2 \times 3 = 6$).

We generate artificially goals for an input data set of $\mathbf{p}_{target}$ with a direct simulation and we will try to find back with the optimization $\mathbf{p}_{target}$ by starting with random values of $\mathbf{p}_{init}$.

We can see in Fig. 2 the simulation setup with the different configurations having all the same common parameters $\mathbf{p}_{init}$. The configurations 2 and 3 are rotated beam along their longitudinal axis to have information about the fiber direction influence. We also add configuration 4 where we attached the beam from the other end to equalize the importance of the mechanical parameters on either side of the beam.

B. Influence of the configurations on the convergence

As explained above the number of parameters, the variety of configurations and the placement of effectors are key. They have a direct link with the optimisation convergence. To illustrate this, we can see on Fig. 3 different optimisation

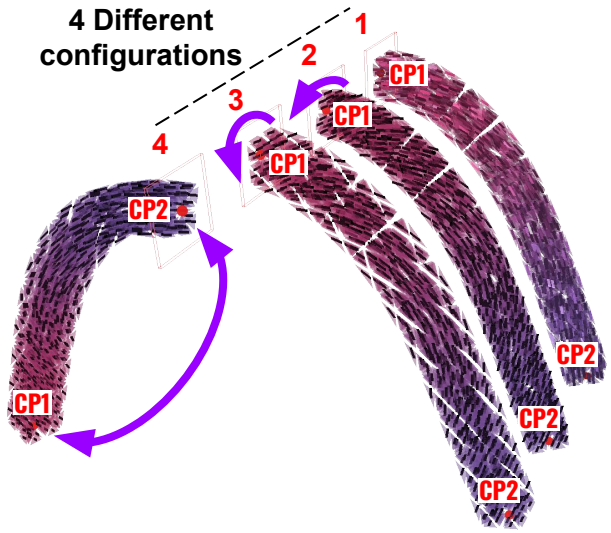

Fig. 2: QP simulation with 4 configurations of a simple beam subject to only gravity. Purple arrows represent the new change in the configuration compared to the first (**1**). Configuration **2** & **3** have two different rotation around there longitudinal axis and for **4** the tip and attached part were switched.

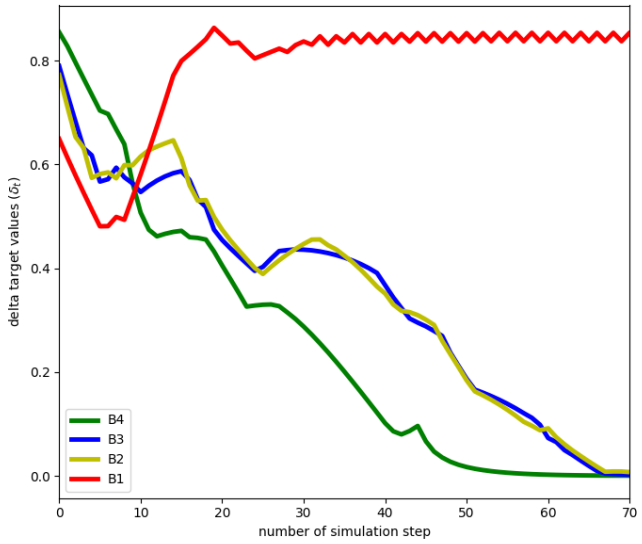

Fig. 3: Minimisation of $\mathbf{\Delta}_e$ in different QP problem. B1 = QP problem with one configuration, B2 with 2 ect...

convergence each with a different number of configurations of the simple beam presented previously (Fig. 2) with only one effector at its tip. When we only have one configuration (red) the QP problem get stuck on a local minima which does not allow to reduce further $\mathbf{\Delta}_e$. With 2 and 3 configurations we get to a $\mathbf{\Delta}_e \approx 0$ but around simulation step 65. The third configuration probably does not provide enough relevant information. In comparison, the fourth configuration, inversed (the beam is now attached by its other end) brings useful information by making "visible" the effect of some parameters. The problem converges more quickly ($\approx$ step 50). Through this example, we can see that it is useful to have mathematical tools to understand if the optimisation problem we are creating is well posed.

Then we test the optimisation when starting with random initial $\mathbf{p}_{init}$. By minimising the beam tip position difference between the input data and the position obtained in the optimisation, we test the convergence and observe if we reach the same $\mathbf{p}_{target}$.

We have launched the optimisation 30 times with, each time, a new randomized $\mathbf{p}_{init}$ and as we can see on Fig. 4 for the parameters E_t and E_l of the 2 *CP*, the QP problem converge each time to the same values which are the right $\mathbf{p}_{target}$. Note that we have chosen a certain range on the parameter space: $[E_{min}, E_{max}]$. Also, for stability issues, we have forced the following inequality between longitudinal and transverse Young modulus : $E_l < E_t$.

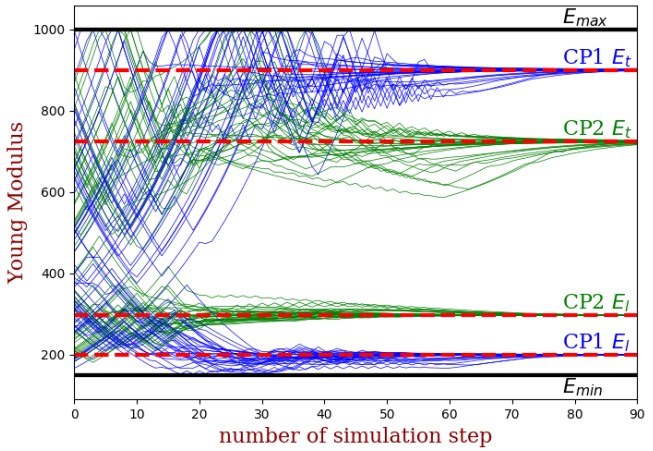

Fig. 4: E_t and E_l convergence from the simulation of Fig. 2 for 30 different starting of $\mathbf{p}_{init}$. Blue/green lines are respectively for CP_1/CP_2 parameters and the red dotted lines correspond to the different parameters objectives. The 2 black lines represent the upper/lower bound (E_{max}/E_{min}) we have put for E.

C. Conditioning analysis on soft anisotropic beams

Through this optimisation example of anisotropic beams, we illustrate in this section how it is possible to study the compliance matrix conditioning, to have a better understanding about the convergence/divergence of the QP problem.

In Eq. (11), the matrix $\mathbb{W}_{ep}$ represents the influence of the mechanical parameters variations on the effectors positions since we have, for each configuration j, $\boldsymbol{\delta}_e^j = \mathbf{W}_{ep}^j \boldsymbol{\lambda}_p^j + \boldsymbol{\delta}_e^{j,0}$. To study conditioning, one can do a singular value decomposition (SVD) of the compliance $\mathbb{W}_{ep} = \mathbf{USV}^T$. The range of values in the matrix $\mathbf{S}$ (between the largest and the smallest) is related to the conditioning value of the problem. If this matrix is poorly conditioned, i.e. with a high condition number, there will be modifications of parameters that will have no effect on the movements of effectors. In practice this will be likely to make the QP optimiser fail.

Number of		configuration		
parameter	effector	3	4	6
4	1	112	8	4
	4	27	5	4
6	1	1413	221	143
	4	84	72	53

TABLE I: Comparison of different conditioning values of $\mathbb{W}_{ep}\left(\mathbb{W}_{pp}\right)^{-1}$ with different number of parameters $\{4, 6\}$ per beam and effector on them $\{1, 4\}$ and finally with 3, 4 or 6 configurations. **Green** correspond to converging cases, **orange** to oscillating and **red** to diverging.

In this situation, we can extract the corresponding kernel vectors (the vectors of $\mathbf{U}$ corresponding to the small singular values, say $\mathbf{U}_s$) that will provide the directions in which modifying the mechanical parameters has almost no influence on the movement of the effector points. In practice, it means the effector points do not change positions using 2 different sets of mechanical parameters $\mathbf{p}$ and a variation of $\mathbf{p}$ along the direction $\mathbf{U}_s$: $\mathbf{p} + \alpha\mathbf{U}_s$.

This information is very rich as it provides parameter directions that have not been sufficiently excited in the input data set of the calibration. This allows to point out which parameter or combination of parameters needs to be particularly exposed in additional configurations, allowing to improve the conditioning.

However, analysing the concrete conditioning values obtained for the matrix $\mathbb{W}_{ep}$ could be difficult as they are also dependent of nominal parameters, such as the global compliance. So a large conditioning value could still sometimes be acceptable. We can reduce this effect by using the square matrix $\mathbb{W}_{pp}$ (similar to $\mathbb{W}_{aa}$ definition) obtained by summing up $\sum^{j} \mathbf{H}_p\mathbf{K}^{-1}\mathbf{H}_p^T$ on each of the j configurations. If we compute $\mathbb{W}_{ep}\left(\mathbb{W}_{pp}\right)^{-1}$ we obtain a dimensional link between the parameters and the effectors for which the conditioning number is easier to interpret.

We can see in Tab. I different results of conditioning for $\mathbb{W}_{ep}\left(\mathbb{W}_{pp}\right)^{-1}$ depending on combinations of number of parameters/effectors/configurations. We can see that when we optimise with less parameters and with more configurations, the conditioning values tends to decrease which improve the convergence. On the contrary, we rapidly tend to divergence cases when we have less configurations with less effectors. These conditioning values are therefore good indicators of the variety of data used for calibration (with respect to the tested parameters) but unfortunately, we cannot yet presume what is the bound from which the problem will diverge/converge.

VI. Practical Usage: Calibration of a Real Anisotropic Parallel Robot

In [15] as we can see in Fig. 5, we have shown that we were able to create a new DOF on a 6-DOF parallel soft robot using metamaterial and simulate this new behavior. We have also done an initial workspace assessment on the simulation. In this previous work, mechanical parameters were fed to the model in a fairly accurate way through theoretical mechanical parameters of the soft plastic used in the 3D-printer as well as numerical homogenisation to take into account the particular foam structure of the meta-material. Nevertheless, we were controlling the robot on open loop and we witnessed large differences between the simulated workspace capability and the reality. This can come from multiple factors/errors :

- Printing technique adding defects into the structure and also adding some anisotropy along the printing direction.
- homogenisation of the generated mesostructure mechanical properties.
- Setup little errors (placement, motors position, mechanical fatigue, noise...).

A calibration of the model based on real data, taken from the robot, should allow to significantly improve the results of the model, in particular by reducing the first two error factors.

A. QP problem setup

As effector, we have chosen the frame placed at the middle of the rod linking the 2 tripods. We have placed on it a Polhemus[3] sensor that allows us to get accurate position as well as orientation of the effector during the calibration. To characterize our soft sheet mechanical properties we choose to put *CP* as shown on Fig. 6: two for each branch (with none at their end because they are rigidified to represent the attached part to the motors) and then 6 around the center where most of the precision is needed. The $\mathbf{p}_{init}$ are given in order to be as close as possible with the previous simulated model (see A of Fig. 6), which means a uniform value of the Young modulus E_t and E_l along the structure and the same spiral pattern parameterized with θ.

Assuming that 3D printing is sufficiently reproducible, we consider that the robotic setup is symmetrical. The same parameters will be used in the two soft sheets. Consequently, we have 12 *CP* with 3 parameters each, so a problem of $dim(\mathbf{p}) = 36$. For the calibration, we choose 7 configurations (which is the minimum to guarantee the positiveness of the QP matrix) giving 7 effector positions well distributed inside the workspace. Each effector position is described with 6 parameters (3 for translations, 3 for orientations) so we have $dim(\mathbf{\Delta}_e) = 42$.

B. Results

The optimisation converged to 36 new values giving to a new visual representation of the results with the interpolation shown on Fig. 6.B. It clearly show the shortcomings of 3D printing with respect to anisotropy and expected mechanical properties: compared to Fig. 6.A $\mathbf{p}_{init}$, E_t and E_l are no more homogeneous over the sheet with blue pockets being less stiff than the red. Moreover, the spiral pattern is much less pronounced.

In contrast, with this new set of mechanical parameters the general error reduction between the goal and effector was

[3]https://polhemus.com/

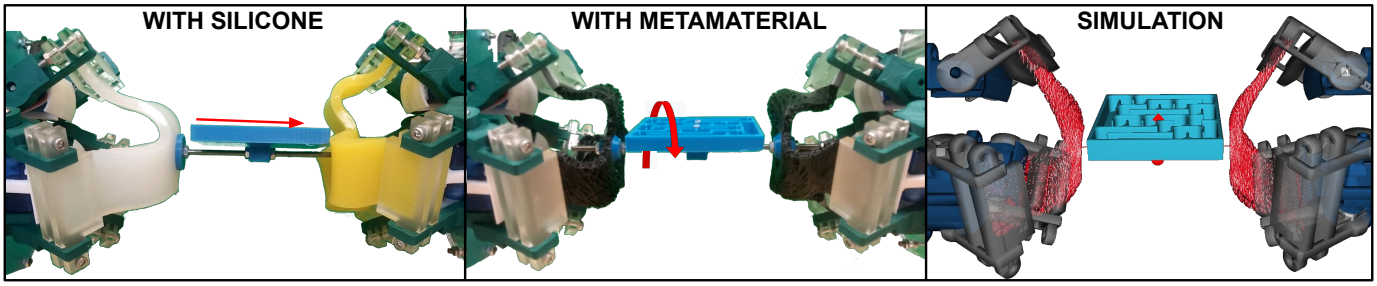

Fig. 5: 6-DOF parallel soft robot of [15], left and middle show that we add a new independent DOF with the metamaterial and right shows that we are able to simulate this new behavior.

reduced by 46% as can be seen on Tab. II. It correspond to a mean error distance of 1.2mm with the previous value and 0.49mm with the new set of parameters found by the optimisation. Concerning the error in angle we pass from $6.35°$to $3.21°$.

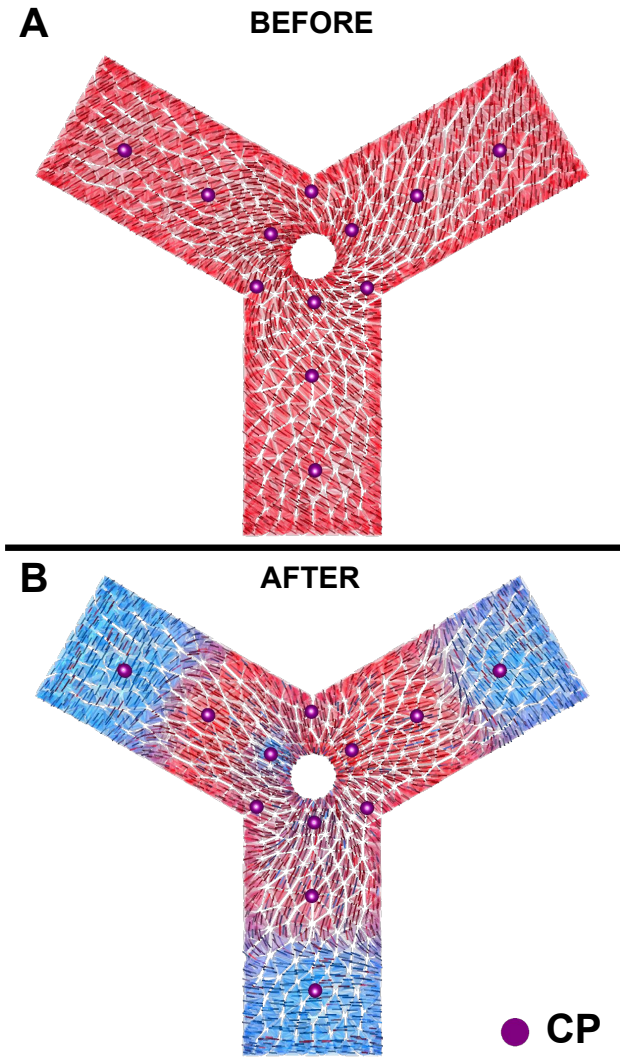

Fig. 6: Flexible sheet before (A) and after (B) optimisation with the *CP* repartition (purple dot).

	configurations (effector position)							mean
	1	2	3	4	5	6	7	
error reduction (%)	60	70	55	47	46	68	-23	46
	new configurations (taken in workspace)							
error reduction (%)	65	65	27	5	17	48	76	43

TABLE II: Errors reduction of the effector position in percent between the real setup and the simulation before and after optimisation. The first line correspond to the error reduction on configuration on which the optimisation was done and the second on newly tested configurations.

To show the validity of the calibration, we also consider new positions in the workspace (that were not used for calibration) and we compare the evolution of the error with the model using the initial (not calibrated) parameters. We can see that it was also reduced in average by 43% which shows the FEM ability to interpolate the behaviour beyond the positions on which it was calibrated.

A part of the remaining error (which is noticeably smaller than before) comes from the fact that the model is not perfect (mechanical parameters comes from homogenisation, internal contact are not taken into account, model for small displacement for linear elastic material...) and another part comes from the real setup noises that are not taken into account in the calibration (positioning error of the motors, dimension and attachment of the sheet with the motors, placement of the rigid marker...). To further decrease the error, more *CP*s could be added to refine the parameters evolution over the robots shape but it will require to calibrate on more configurations.

Nevertheless, the QP optimiser was able to find the best parameters given the input data. The improved model parameters can be used for more accurate control of the robot, including when looking into closed-loop control.

VII. DISCUSSION

We have presented a new method allowing to optimise mechanical parameters of a FEM as well as to formulate and solve multi-configurations QP problems. The method allows to calibrate a complex anisotropic FEM with a relatively high number of parameters (here up to 36) while using a relatively low number of different configurations (only 7), with a single

6D-measurement per configuration, and succeeds to achieve an average error reduction, before/after calibration, of more than 40%. The number of input data remains reasonable. The method is frugal compared to model-less methods.

The method is based on convex optimisation and we have shown that the combination of the number of configurations, the number of parameters, and the number of effectors have to be carefully chosen to guarantee the well-posedness of the QP problem. In section V-C, we give an indicator to evaluate the conditioning of the QP problem. However that is not a strict criteria that will predict whether the optimisation will converge or not.

But another aspect of the problem must be mentioned: convex optimisation can lead to getting stuck on local minima, in particular when the choice of parameters or the stresses in terms of deformations lead to important non-linearities where it is difficult to reach the global optimum. Moreover, we rely on numerical differentiation that assumes a good continuity of the influence of the parameters on the expression of the internal forces.

We have discussed ways to set the optimisation problem properly but it still needs some investigation in order to have a better knowledge about the problem conditioning and as much as possible, we would like to provide an automatic calibration workflow. We could test the method on various soft robots, with other actuator types and also other sensors to make the calibration.

Also we have to keep in mind that the optimisation process over real life data will try to correct the total error of our real setup (motors actuation / mounting errors...) and not only the mechanical parameters of the soft structure. It would need more effort to have a more reliant setup in order to decouple the mechanical errors specifically. Note that we are doing open loop simulation for the motors control, closing the loop and sending back more sensors informations (like real motor positions and/or torques) can help achieve this mechanical decoupling.

VIII. CONCLUSIONS AND PERSPECTIVES

The calibration method based on QP optimisation presented in this study is generic and can be applied to any geometry of soft robot as well as any FEM with other constitutive law (in principle). In this letter, we show that it is applicable in the complex case of an anisotropic mechanical model, which involves "traditional" mechanical parameters like Young modulus, but also more specific parameters like directions of anisotropy. We have demonstrated that the method can handle this complexity and is able to find satisfying optimums as long as enough informative configurations are provided.

This method could also be used for design purposes by combining mechanical parameters optimisation with actuation in order to have a robot structurally optimised for a specific task. This will be a direction for future works.

REFERENCES

[1] Scott Kuindersma, Frank Permenter, and Russ Tedrake. "An efficiently solvable quadratic program for stabilizing dynamic locomotion". In: ICRA 2014. IEEE. 2014.

[2] HeeSun Choi et al. "On the use of simulation in robotics: Opportunities, challenges, and suggestions for moving forward". In: Proceedings of the National Academy of Sciences (2021).

[3] James M Bern et al. "Soft robot control with a learned differentiable model". In: RoboSoft 2020. IEEE. 2020.

[4] Sam Kriegman et al. "Scalable sim-to-real transfer of soft robot designs". In: RoboSoft 2020. IEEE. 2020.

[5] Yuanming Hu et al. "Chainqueen: A real-time differentiable physical simulator for soft robotics". In: ICRA 2019. IEEE. 2019.

[6] Javier Tapia et al. "Makesense: Automated sensor design for proprioceptive soft robots". In: Soft robotics (2020).

[7] Christian Schumacher, Espen Knoop, and Moritz Bächer. "Simulation-ready characterization of soft robotic materials". In: IEEE Robotics and Automation Letters (2020).

[8] S Connolly, Donald MacKenzie, and Tugrul Comlekci. "Multi-objective optimization of hyperelastic material constants: A feasibility study". In: Constitutive Models for Rubber X. CRC Press, 2017.

[9] Moritz Bächer, Espen Knoop, and Christian Schumacher. "Design and Control of Soft Robots Using Differentiable Simulation". In: Current Robotics Reports (2021).

[10] Jonàs Martınez et al. "Polyhedral Voronoi diagrams for additive manufacturing". In: ACM Transactions on Graphics (TOG) (2018).

[11] Félix Vanneste et al. "Anisotropic soft robots based on 3D printed meso-structured materials: design, modeling by homogenization and simulation". In: IEEE Robotics and Automation Letters (2020).

[12] Eulalie Coevoet et al. "Software toolkit for modeling, simulation, and control of soft robots". In: Advanced Robotics (2017).

[13] Yijing Li and Jernej Barbič. "Stable anisotropic materials". In: IEEE TVCG (2015).

[14] Donald Shepard. "A two-dimensional interpolation function for irregularly-spaced data". In: Proceedings of the 1968 23rd ACM. 1968.

[15] Félix Vanneste, Olivier Goury, and Christian Duriez. "Enabling the control of a new degree of freedom by using anisotropic material on a 6-DOF parallel soft robot". In: Robosoft 2021. 2021.