IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2024, Yokohama, Japan. Cite as RA-L paper.

Optimal Design of Flexible-Link Mechanisms with Desired Load-Displacement Profiles

Guirec Maloisel[1,2], Espen Knoop[1], Christian Schumacher[1], Bernhard Thomaszewski[2], Moritz Bächer[1], Stelian Coros[2]

***Abstract*—Robot mechanisms that exploit compliance can perform complex tasks under uncertainty using simple control strategies, but it remains difficult to design mechanisms with a desired embodied intelligence. In this paper, we propose an automated design technique that optimizes the desired load-displacement behavior of planar flexible-link mechanisms. To do so, we replace a subset of rigid with flexible links in an existing mechanism, and optimize their rest shape. We demonstrate the efficacy of our approach on a set of examples, including two fabricated prototypes, illustrating applications for grasping and locomotion tasks.**

***Index Terms*—Soft Robot Materials and Design; Compliant Joints and Mechanisms; Optimization and Optimal Control**

I. Introduction

COMPLIANT mechanisms bear the potential to build robots that can locomote in varied and uncertain terrain, or manipulators that can pick up vastly different objects using a single, simple control strategy. However, it remains challenging to design mechanisms that have a desired embodied intelligence [1], meaning that the right amount of compliance is introduced so that a fixed control strategy remains functional for a user-specified range of interaction-induced forces or displacements.

We propose to tackle this by starting from a user-specified rigid mechanism and a simple control strategy, replacing a subset of rigid with flexible links and optimizing their rest shape while keeping the control strategy fixed.

Because we introduce flexibility in links rather than joints (as is commonly done in compliant mechanism design), our flexible mechanisms maintain the same kinematic behavior as the input mechanism if no forces are present. This means that we can utilize the design space of the flexible links to generate a user-desired local behavior to, for example, achieve robustness under an uncertain behavior.

To define the desired behavior, we propose to use user-specified samples of load-displacement curves at a set of locations of expected interactions, in defined directions. This is a good interface for the following reasons: a load-displacement curve is straightforward to understand, yet non-restrictive because the behavior in several directions at a single, as well as at several points, can be specified. By analyzing the behavior of the initial design under representative loads, we can visualize the initial load-displacement curves and provide the user with a starting point for "editing" the mechanism's behavior. As such, the curves also serve as proxy for the more abstract embodied intelligence of the mechanism.

We then solve a shape optimization problem for the flexible links, which seeks to match the prescribed force-displacement pairs as closely as possible.

Succinctly, our technical contributions are:

- A *differentiable quasi-static simulator* for flexible-link mechanisms, enforcing the coupling between flexible-flexible and flexible-rigid link pairs, as also mechanical joints, with constraints in a Lagrangian formulation.
- A *design optimization* that achieves a desired load-displacement profile at user-specified points of interest, leveraging sensitivity analysis to compute derivatives of simulation states with respect to design parameters.
- A *smooth remeshing-free parameterization* of the volumetric rest shape of flexible links using splines and an extension of the Bounded Biharmonic Weights technique by Jacobson et al. [2].
- A set of demonstrations, including two fabricated examples, that illustrate the utility of our technique in the automated design of manipulators and soft robotic components with applications in grasping and locomotion.

To the best of our knowledge, we are the first to propose a system to design planar mechanisms with embodied intelligence, interfacing with existing CAD models and an easy-to-understand interface to specify a desired behavior in a local neighborhood of a mechanism performing a control strategy.

II. Related Work

Compliant Mechanism Design: In compliant mechanism design, we differentiate between mechanisms that are fully compliant, or have their compliance concentrated at joints [3].

For the latter, designs are typically obtained with Rigid Body Replacement methods [4], substituting flexures for joints. Specific simulation or optimization methods include the Pseudo-Rigid-Body model [5], the Freedom and Constraint Topologies method [6], or a rod-based technique that optimizes the shape of flexures after joint substitution [7]. While our

Manuscript received: October 14, 2022; Revised: March 16, 2023; Accepted: May 1, 2023.

This paper was recommended for publication by Editor Yong-Lae Park upon evaluation of the Associate Editor and Reviewers' comments. This work was supported in part by the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme under Grant agreement No. 866480.

[1]Guirec Maloisel, Espen Knoop, Christian Schumacher and Moritz Bächer are with Disney Research, Stampfenbachstrasse 48, 8006 Zurich, Switzerland. first.lastname@disneyresearch.com

[2]Guirec Maloisel, Bernhard Thomaszewski and Stelian Coros are with Department of Computer Science, ETH Zurich, Switzerland. bthomasz@inf.ethz.ch, scoros@inf.ethz.ch

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

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2024, Yokohama, Japan. Cite as RA-L paper.

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2024, Yokohama, Japan. Cite as RA-L paper.

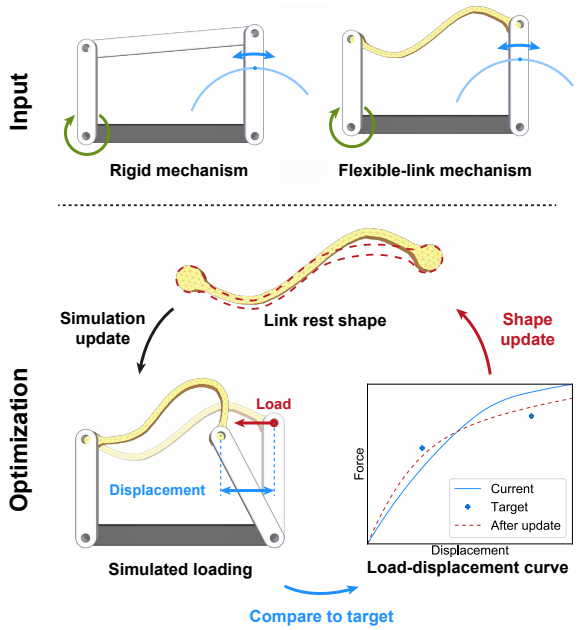

Fig. 1. **Overview**. Input mechanism with rigid links (top, left) and a kinematically equivalent, compliant version where a subset of links is replaced with flexible components (top, right). At each iteration, the simulator updates the mechanism state under prescribed external loads, given the current link design (bottom, left). The optimizer then computes the error with respect to target force-displacement pairs, and updates the rest shape of links using derivatives of the state with respect to design parameters (bottom, right).

design optimization shares similarities with Megaro et al. [7], we propose to replace rigid with flexible links instead of substituting flexures for joints.

Shape and topology optimization techniques are common tools to design mechanisms with distributed compliance [8], avoiding stress concentrations at pseudo-hinge joints [9]. However, because we preserve mechanical joints, these techniques are not directly applicable to our design problem.

Flexible-Link Mechanisms: Mechanisms that combine mechanical joints with flexible links have received limited attention, pioneered by Burns' work on four-bar linkages with one flexible link [10]. Work on their classification [11], analysis [12], and synthesis [13] followed. However, only flexible linkages with a zero-mobility rigid equivalent were considered. In [10], flexible links with a pin joint at both ends are deemed of little interest, because they do not differ kinematically from a rigid link. In contrast, we propose to make use of this kinematic equivalence, utilizing the design space for optimizing a mechanism's embodied intelligence rather than having to approximate a rigid mechanism's kinematics first.

Optimal Load-Displacement Behavior: Previous work has explored the design of multistable [14], constant-force [15], statically balanced [16, 17] mechanisms and nonlinear springs [18]. In contrast, we propose a technique that enables the optimization of a desired load-displacement behavior of planar mechanisms given a simple, user-defined control strategy. While specific designs of compliant feet [19, 20] or grippers [21, 22, 23] exist, we automate their design.

III. Overview

Our technique takes a functional rigid mechanism as input (see Fig. 1 top, left). Even though our simulation representation is general, we focus on planar mechanisms in this paper. To avoid out-of-plane deformations, we set the thickness of flexible links to a sufficiently large value. The user then selects a subset of links that should be made compliant, and assigns a material to them (top, right). We assume that the links are sufficiently stiff for a quasi-static model to be sufficient. Under this assumption, the kinematic behavior of the flexible-link mechanism is preserved if no external or body forces are acting on the components. The only constraint is that the joint locations need to be preserved.

The response of the mechanism to external perturbations is now a function of the rest shape of the flexible links. Using a differentiable simulation (Fig. 1 bottom, left; Sec. IV), we predict the load-displacement behavior at a single or at several configurations of interest. Under configuration we understand a set of position values that remain unchanged during optimization. We therefore keep the control policy fixed, and optimize the mechanism's embodied intelligence only.

With analytical derivatives of simulation states with respect to design parameters, we can then iteratively optimize the load-displacement behavior (Fig. 1 bottom, right; Sec. V) of, for example, a gripper in its closed configuration (Sec. VII). The user specifies a target load-displacement behavior for each configuration of interest with a set of load-displacement sample points that we use for optimization.

IV. Simulating Flexible-Link Mechanisms

To simulate our flexible-link mechanisms, we rely on a quasi-static model. We set the motors to a prescribed position, and solve for the static equilibrium of rigid and flexible components under mechanical joint constraints and external forces

$$\min_{\mathbf{s}} E(\mathbf{s}) \;\; \text{s.t. } \boldsymbol{\mathcal{C}}(\mathbf{s}) = \mathbf{0} \tag{1}$$

where $\mathbf{s}$ is the state of the mechanism, E the total potential energy, and $\boldsymbol{\mathcal{C}}$ the vector of kinematic constraints. The energy E sums up the internal energy of all deformed flexible components, subtracting the work that external forces perform on the mechanism. We will discuss the state vector, the energy, and the constraints in more detail below.

The constrained minimization problem (1) is solved by forming the Lagrangian

$$\mathcal{L}(\mathbf{s}, \boldsymbol{\lambda}) = E(\mathbf{s}) - \boldsymbol{\lambda}^T \boldsymbol{\mathcal{C}}(\mathbf{s}) \tag{2}$$

with Lagrange multipliers $\boldsymbol{\lambda}$. In our implementation, we use a standard Augmented Lagrangian Method [24].

Note that this formulation is general, and is not restricted to planar nor to the specific flexible-link mechanisms we consider here. As such, our simulator can handle arbitrary flexible multibody systems that consist of rigid and flexible bodies whose relative motion is restricted by mechanical joints of varying degrees of freedom. Moreover, it could be used to check if there is out-of-plane buckling behavior.

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2024, Yokohama, Japan. Cite as RA-L paper.

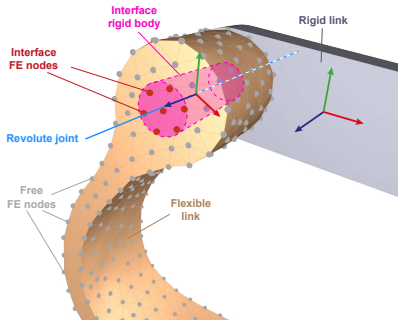

Fig. 2. **Mechanism State Representation.** The FE nodes within a volume that is defined by a joint's degrees of freedom (in red) collectively form an interface rigid body that enables us to formulate kinematic constraints between flexible and rigid links.

A. Mechanism State Representation

The state vector, $\mathbf{s}$, is formed by the concatenation of the degrees of freedoms (DoFs) of the different rigid and flexible links. Rigid links contribute 6 DoFs (3D position and orientation). In practice, we formulate rigid body rotations as unit quaternions, as done in [25], so that rigid links are parameterized by a 7-vector. A corresponding constraint that enforces the unit length of quaternions is appended to the vector $\boldsymbol{\mathcal{C}}$ [25].

The degrees of freedom of flexible links are defined by their finite-element (FE) discretization (3 DoFs per FE node), and also appended to vector $\mathbf{s}$. To model the behavior of mechanical joints, such as, e.g., a hinge, we define *interface rigid bodies* by collecting all FE nodes within a volume that is defined by the DoFs of a joint (see Fig. 2). This set of nodes is constrained to move rigidly, and is used to implement mechanical joint constraints between neighboring pairs of flexible and rigid bodies, and pairs of flexible bodies. We then omit these FE nodes, adding a 7-vector for the DoFs of the interface rigid body to $\mathbf{s}$ instead.

B. Mechanical Joint Constraints

To formulate kinematic constraints, $\boldsymbol{\mathcal{C}}$, for joints and actuators, we use the technique described by Schumacher et al. [25]. To formulate constraints between pairs of flexible-flexible and flexible-rigid links, we introduce constraints that restrict the relative motion of the states of the respective interface and standard rigid bodies.

C. Total Potential Energy

The total potential energy consists of two terms

$$E = E_{\text{int}} - E_{\text{ext}}. \tag{3}$$

The elastic energy, E_{int}, sums up the internal energy of all the deformed finite elements of the flexible links, numerically integrating a standard strain energy density [26] over the elements' volumes. For the materials we use in our demonstrations, a neo-Hookean model is sufficient.

Note that interface rigid bodies, while viewed as rigid bodies in the formulation of the state and constraints, consist of FE nodes in order to implement the coupling between rigid and flexible bodies, and therefore also contribute to E_{int}.

From the elastic energy, we subtract the work E_{ext} of external forces by summing up 3D forces, $\mathbf{f}_f$, that act at fixed positions, $\mathbf{p}_f(\mathbf{s})$, on the mechanism, times the displacement

$$E_{\text{ext}} = \sum_{\text{forces } f} \mathbf{f}_f^T \mathbf{p}_f(\mathbf{s}), \tag{4}$$

omitting the constant term (force times initial position). To compute the position $\mathbf{p}_f$, we apply a rigid body transformation to a fixed location in rigid body coordinates, or an FE interpolation for fixed positions on flexible bodies.

V. Optimizing the Load-Displacement Behavior

The input to our design optimization is a set of configurations, which are uniquely defined by a set of position values for all actuators, that remain fixed during optimizations. For each configuration, the user specifies load-displacement samples for points of interest on the mechanism. Setting the actuators to a particular configuration i, and the forces to the user-specified values for sample j, we evaluate the performance of the current design $\mathbf{p}$ by comparing simulated displacements $\mathbf{d}_{ij}$ to specified displacements $\bar{\mathbf{d}}_{ij}$

$$\mathcal{O}(\mathbf{p}) = \sum_{\text{config. } i} \sum_{\text{sample } j} \frac{1}{2} \|\mathbf{d}_{ij}(\mathbf{s}_{ij}(\mathbf{p})) - \bar{\mathbf{d}}_{ij}\|^2. \tag{5}$$

We only evaluate the performance of a mechanism design at a sparse set of configurations and force-displacement samples. This is sufficient as long as we can assume that the quasi-static state of a mechanism is not path-dependent.

A. Regularization

To preserve the accuracy of our simulator and avoid numerical stiffening, we must ensure that the mesh quality of the flexible components remains sufficiently high. To this end, we add to our objective $\mathcal{O}$ the regularization term

$$\mathcal{R}_{\text{mesh}}(\mathbf{p}) = \sum_{\text{element } e} \beta\left(\rho(e)\right) \tag{6}$$

penalizing elements with low-quality rest shape. ρ measures the aspect ratio of an element, and β is a $\mathcal{C}^2$ barrier function (see Appendix) that prevents too large aspect ratios. For tetrahedral elements, the ratio between the circumradius and the radius of an inscribed sphere [27] is a measure with desirable properties: among alternatives (see, e.g., [28] for an overview), this metric has the advantage of being differentiable (no $\min$ or $\max$ operator) with respect to the vertices of a tetrahedron. Furthermore, we monitor element inversions (elements with negative volume) at all optimization steps, backtracking if we detect any by setting $\mathcal{R}_{\text{mesh}}$ to infinity.

As we discuss in more detail in the next section, we use a spline-based parameterization, defining the centerline of links and the local thickness. To limit the search space to non-degenerate link shapes, and to ensure fabricability, we introduce an additional regularization term $\mathcal{R}_{\text{shape}}$. This second regularizer prevents links from crossing a maximal curvature bound, and from becoming too thin or too thick. To bound

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2024, Yokohama, Japan. Cite as RA-L paper.

the local curvature, we sample the spline and bound the ratio, $r = \frac{h}{R}$, for every three consecutive samples, where h is the average segment length between samples and R is the circumradius of the three samples ($R = \infty$, and $r = 0$ for samples on a line). To bound the in-plane thickness along the spline, we penalize the interpolated thickness at sample points. To implement these penalties, we use the same barrier function as in our mesh quality regularizer (see Appendix).

Finally, to ensure that the design problem is bounded, we ask the design parameters to remain close to their initial values $\mathbf{p}^{\text{init}}$

$$\mathcal{R}_{\text{init}}(\mathbf{p}) = \frac{1}{2}\left\|\mathbf{p} - \mathbf{p}^{\text{init}}\right\|^2. \tag{7}$$

B. Optimization

To solve this problem using gradient-based optimization, we need to take derivatives of the implicit function, $\mathbf{s}(\mathbf{p})$, defined by the force equilibrium equation, or equivalently by first-order optimality of the forward simulation

$$\frac{\partial \mathcal{L}}{\partial \mathbf{y}}(\mathbf{y}, \mathbf{p}) = \mathbf{0} \tag{8}$$

in which $\mathbf{y}$ refers to the concatenation $\mathbf{s} \mid \boldsymbol{\lambda}$. To do so, we rely on sensitivity analysis [29]: applying the implicit function theorem to Eq. 8 yields the derivatives

$$\frac{\mathrm{d}\mathbf{y}}{\mathrm{d}\mathbf{p}} = -\left(\frac{\partial^2 \mathcal{L}}{\partial \mathbf{y}^2}\right)^{-1} \frac{\partial^2 \mathcal{L}}{\partial \mathbf{y} \partial \mathbf{p}} \tag{9}$$

from which we extract the desired $\frac{d\mathbf{s}}{d\mathbf{p}}$. By applying the chain rule, we obtain the objective gradient

$$\frac{\mathrm{d}\mathcal{O}}{\mathrm{d}\mathbf{p}} = \sum_{i,j} \frac{\partial \mathcal{O}_{ij}}{\partial \mathbf{s}_{ij}} \frac{\mathrm{d}\mathbf{s}_{ij}}{\mathrm{d}\mathbf{p}} + \frac{\mathrm{d}\mathcal{R}_{\text{mesh}}}{\mathrm{d}\mathbf{p}} + \frac{\mathrm{d}\mathcal{R}_{\text{shape}}}{\mathrm{d}\mathbf{p}} + \frac{\mathrm{d}\mathcal{R}_{\text{init}}}{\mathrm{d}\mathbf{p}}$$

with objectives $\mathcal{O}_{ij} = \frac{1}{2}\|\mathbf{d}_{ij}(\mathbf{s}_{ij}(\mathbf{p})) - \bar{\mathbf{d}}_{ij}\|^2$.

We can then compute the optimal design parameters using standard quasi-Newton (BFGS) minimization [24].

VI. Remeshing-Free Shape Parameterization

In this section, we describe how we parameterize the rest shape of flexible links, providing us with design parameters $\mathbf{p}$ for optimization. We focus our discussion on a single link, and formulate a parameterization that fulfills the following requirements:

- *Differentiability*: Changes to the connectivity of a volumetric mesh are discontinuous operations, and therefore ill-suited for design optimization. We therefore require our parameterization to preserve the connectivity.
- *Mesh Quality*: A candidate parameterization needs to keep the quality of elements sufficiently high for a wide range of design parameters.
- *Joint Interface Preservation:* Nodes that represent the interface to a neighboring rigid or flexible link need to remain unchanged, except for undergoing a rigid body transformation that the degrees of freedom of the joint allow.

Because links are relatively slender, splines are a natural choice. A small number of cubic Hermite spline segments, routed along the link centerline, provide sufficient control of a link's shape while keeping the parameter count small. However, it is non-trivial to map changes of curves to changes of a volumetric mesh while also fulfilling the above list of requirements. To define this mapping, we rely on Bounded Biharmonic Weights (BBW) [2]. We first define a set of handles whose degrees of freedom are controlled by splines. These handles then propagate the design changes to the mesh nodes of a link's rest configuration. A key advantage of using BBW is the C^∞-smoothness of the mapping as long as mesh nodes are not co-located with handles. Moreover, BBW interfaces with a versatile set of handle types that enables the specification of custom parameterizations. This could help to extend our parameterization to optimize spatial mechanisms.

A. Spline-based Parameterization

We assume that each flexible link is volumetrically meshed in the mechanism's rest pose. To define our parameterization, we initialize a set of cubic Hermite spline segments (in blue in Fig. 3 top) that we route along the centerline. To do so, we co-locate a control point with each mechanical joint, and place additional control points on the approximate centerline. In-plane tangent vectors are initialized so that the spline representation closely approximates the centerline.

While 2D splines with two position and two tangent parameters per control point suffice for a parameterization of a link's centerline, they do not allow us to vary the local thickness of a link. To parameterize the thickness, we use the z-components of the position and tangent vector of *3D-spline* control points. We initialize the local thickness to 1 and the corresponding tangent vector component to 0 (for all control points). A value of 1 means that the local thickness is unchanged.

The design parameters $\mathbf{p}$ that we change during optimizations are 3D positions and 3D tangents of all control points, excluding the 3D position parameters of control points at mechanical joints to preserve their rest configuration position and the local thickness. The mapping of design parameters to an interpolated position and tangent at the location t along the spline is therefore

$$\mathbf{p}, t \mapsto \begin{bmatrix} \mathbf{x}(\mathbf{p}, t) \\ \delta(\mathbf{p}, t) \end{bmatrix}, \quad \text{and} \quad \mathbf{p}, t \mapsto \begin{bmatrix} \mathbf{t}(\mathbf{p}, t) \\ \dot{\delta}(\mathbf{p}, t) \end{bmatrix}, \tag{10}$$

where δ is the local thickness at point $\mathbf{x} \in \mathbb{R}^2$, and $\mathbf{t} \in \mathbb{R}^2$ the corresponding tangent. Because we set the local thickness and its derivative, $\dot{\delta}$, to 1 and 0 in $\mathbf{p}^{\text{init}}$, the local thickness is 1 (i.e., unchanged), everywhere along the spline at the start of a design optimization.

B. Propagation of Design Changes to Simulation Mesh

To propagate changes of design parameters to a link's simulation mesh, we uniformly sample the spline, creating *joint handles* at mechanical joints and *intermediary handles* at regular arc-length intervals along the spline (dark blue rectangles in Fig. 3 middle). We then compute BBW weights w_{ij} that define the mapping of mesh nodes $\mathbf{v}_i^{\text{init}} \in \mathbb{R}^3$ to their new rest pose position $\mathbf{v}_i \in \mathbb{R}^3$

$$\mathbf{v}_i = \sum_{\text{handle } j} w_{ij} \boldsymbol{\mathcal{T}}(\mathbf{p}, t_j; \mathbf{v}_i^{\text{init}}) \tag{11}$$

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2024, Yokohama, Japan. Cite as RA-L paper.

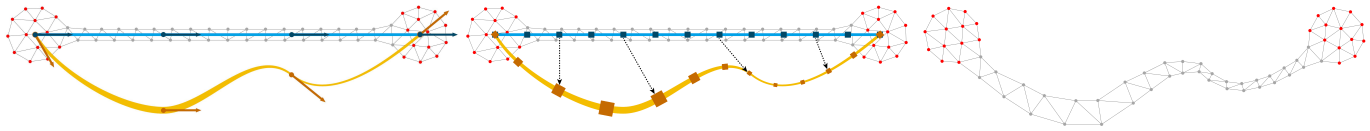

Fig. 3. **Spline-based Parameterization.** Cubic Hermite spline segments (blue, left) are initialized along the centerline of the volumetric link mesh (shown in 2D for simplicity). Design optimizations then change their control points (yellow, left) that we propagate using BBW handles (middle) that we distribute uniformly along the spline, resulting in a link mesh with high-quality elements (right). Constrained regions (red) can only undergo rigid body transformations that the mechanical joint DoFs allow.

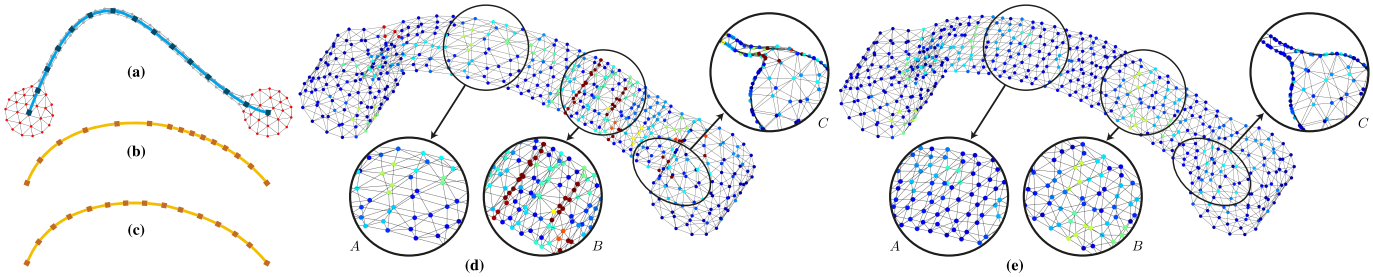

Fig. 4. **Optimizing Handle Transformations. (a)** Side view of the input mesh, spline and handles (blue), with constrained vertices in red. **(b)** Naive handle mapping onto the target spline (yellow), with $t_j = t_j^{\text{init}}$, and α_j prescribed by the spline tangents of joint handles. **(c)** Optimized mapping. **(d, e)** Resulting deformed meshes (d: naive, e: optimized). Vertex colors indicate mesh quality (blue = good, red = bad). Letting intermediary handles "slide" along the spline avoids undesirable stretching (A) or compression (B). Letting joint handles rotate yields smoother transitions with the middle section (C).

by transforming their initial homogeneous coordinates with a 3x4 affine transformation matrix

$$\boldsymbol{\mathcal{T}}(\mathbf{p}, t_j; \mathbf{v}_i^{\text{init}}) = \begin{bmatrix} \delta_j \mathbf{R}_j (\mathbf{R}_j^{\text{init}})^T & \mathbf{0} & \mathbf{x}_j - \mathbf{x}_j^{\text{init}} \\ \mathbf{0}^T & 1 & 0 \end{bmatrix} \begin{bmatrix} \mathbf{v}_i^{\text{init}} \\ 1 \end{bmatrix}$$

where the initial and current 2D positions and rotations, $\mathbf{R}$, of handle j are evaluated at $(\mathbf{p}^{\text{init}}, t_j^{\text{init}})$ and $(\mathbf{p}, t_j)$, respectively.

The weights are computed as in [2], with the following extensions:

- The interface nodes around a joint are constrained to share equal weights.
- As BBW handles must be co-located with mesh vertices, we expand the mesh so it contains all handles, and disregard these new vertices when extracting the deformed shape.
- Besides the Laplacian energy, we minimize the squared out-of-plane gradient of the weights to keep the deformation as-uniform-as-possible along a link's depth.

C. Optimizing Handle Transformations

While it is natural to keep handles at the same locations on a spline, or in other words set $t_j = t_j^{\text{init}}$, this choice can lead to many low-quality elements that restrict the range of design parameters as we can see in Fig. 4 (d). An effective strategy to mitigate this problem is to let joint handles rotate and intermediary handles "slide" along the spline. To do so, we solve for either an angle parameter α_j that defines the 2D rotation $\mathbf{R}_j$ at a joint *or* the location t_j along the spline (intermediary handle) that minimize the aspect ratio ρ of all elements after updating the design parameters $\mathbf{p}$

$$\min_{\boldsymbol{\theta}} \sum_{\text{elements } e} \rho\left(\boldsymbol{\theta}, \mathbf{p}; e\right) \tag{12}$$

with variable vector entries $\theta_j = \alpha_j$ or t_j as appropriate (one variable per handle).

Before evaluating the objective or objective gradient of our design optimization for a new set of parameters, we solve the above problem to identify the optimal handle locations. We then compute the total derivative of FE DoFs in the state vector $\mathbf{s}$ by applying the chain rule

$$\frac{\mathrm{d}\mathbf{v}_i}{\mathrm{d}\mathbf{p}} = \sum_{\text{handle } j} w_{ij} \left(\frac{\partial \boldsymbol{\mathcal{T}}_j}{\partial \mathbf{p}} + \frac{\partial \boldsymbol{\mathcal{T}}_j}{\partial \theta_j} \frac{\mathrm{d}\theta_j}{\mathrm{d}\mathbf{p}} \right) \tag{13}$$

in which $\frac{\mathrm{d}\theta_j}{\mathrm{d}\mathbf{p}}$ is given by the implicit function theorem

$$\frac{\mathrm{d}\boldsymbol{\theta}}{\mathrm{d}\mathbf{p}} = -\left(\frac{\partial^2 \left(\sum_e \rho(e)\right)}{\partial \boldsymbol{\theta}^2} \right)^{-1} \frac{\partial^2 \left(\sum_e \rho(e)\right)}{\partial \boldsymbol{\theta} \partial \mathbf{p}}. \tag{14}$$

VII. Results

We now present several applications of our method to grasping and locomotion problems. In each case, our optimizer was able to produce a design satisfying the user requirements, which were set using simple heuristics according to the problem at hand. To illustrate the transfer of our simulations to the real world, we fabricated and tested two prototypes. Captured and simulated animations of our examples can be seen in the accompanying video.

A. Fabrication

Our flexible links were fabricated in one of two materials across our examples: Smooth-Sil 950 silicone, using injection molding into 3D-printed molds, and Onyx (carbon-filled nylon) links, using a Markforged X7 printer. Onyx parts were left to stabilize for 48 h before use, to ensure consistent material properties.

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2024, Yokohama, Japan. Cite as RA-L paper.

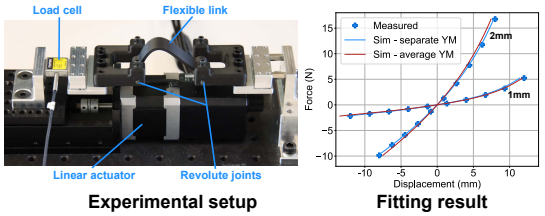

Fig. 5. **Characterization.** Young's modulus fitting for the Onyx material for two sample links, with 1 mm and 2 mm in-plane thickness. We plot simulated forces with Young's modulus set as resulting from the separate fit for each sample (blue curves; 1.89 Gpa / 1.65 GPa for the 1 mm / 2 mm sample), and as per the final average (red curves; 1.77 GPa for both).

Rigid links were 3D-printed in Onyx with a significantly larger in-plane thickness. For revolute joints we used 3mm pins and press-fitted polymer bearings.

B. Simulation

Flexible links were discretized with standard quadratic Lagrange tetrahedral elements, and modeled with a hyperelastic neo-Hookean material. Poisson's ratio and Young's modulus were set to 0.475 and 1.90 MPa for silicone, and 0.42 and 1.77 GPa for Onyx, respectively. The value for Young's modulus was calibrated with a characterization experiment, as we illustrate in Fig. 5 for Onyx. More specifically, our optimizer was used as described in Sec. V, but replacing shape design variables with Young's modulus, to match force-displacement pairs in the captured data. The optimal moduli obtained for material samples of different thicknesses were then averaged.

C. Compliant Claw Gripper

As a first demonstration, we optimize an Onyx compliant claw-like gripper, closing on an object with a simple open-loop controller. The goal is to provide robust grasping over a range of object sizes. More specifically, the grasping force should remain within user-specified minimum and maximum bounds, for a prescribed range of object sizes. This translates into two targets for our optimizer: the displacement of the end-effector perpendicularly to the object (in a closed configuration) must match the smallest or largest object size (5 and 20 mm here), when the minimal or maximal force is applied (4 and 10 N here). For a symmetric design, we run our pipeline on half of the gripper only.

As shown in Fig. 6, the load-displacement profile matches the targets on the optimized design. Note the non-linearity of the resulting profile, which could not be obtained with a linear spring in series with the end-effector. Note also that, in this example as well as others, the compliance is obtained using only the mechanical structure, the end-effector being made rigid to better identify the contribution of our pipeline. In practice, this could be combined with a soft finger pad for better shape adaptation.

We also fabricated a prototype (half-)gripper, on which we captured force-displacement data, using a setting similar to the material characterization experiments (Sec. VII-B; see also our video). For simplicity, the gripper was directly fabricated in

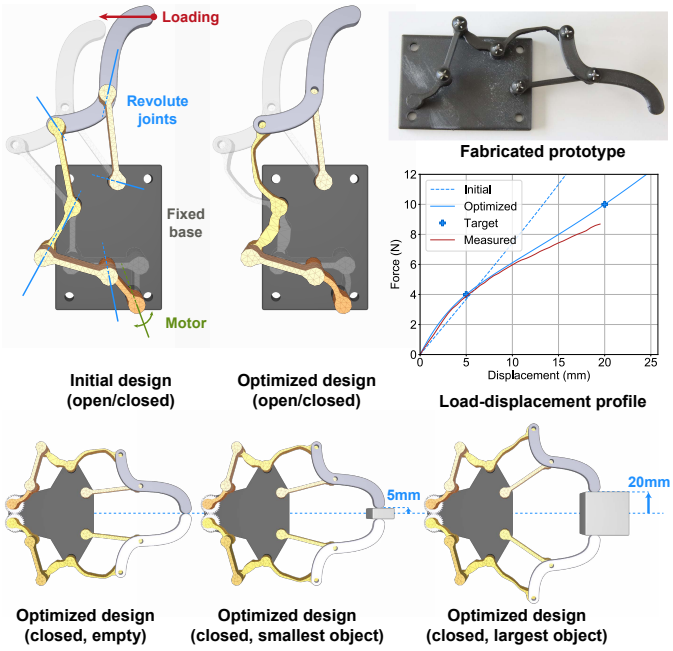

Fig. 6. **Compliant claw gripper**. The base (dark gray) and end-effector (light gray) are rigid, while the 4 other links (orange) are flexible and constitute the design space. The load-displacement curves are computed using our simulator on the initial and optimized design (blue curves), and measured on the fabricated prototype (red curve).

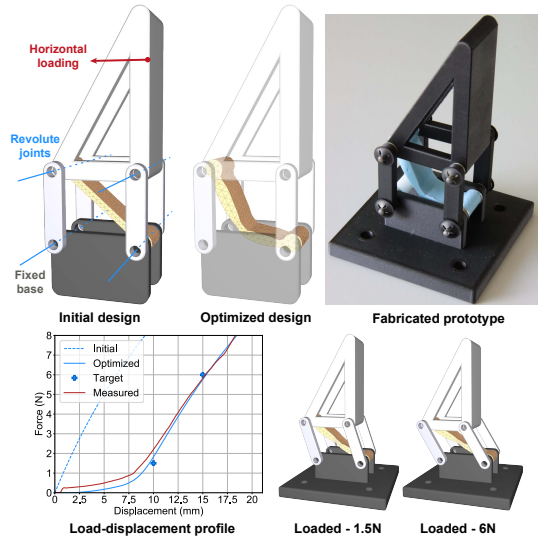

Fig. 7. **Compliant truss gripper finger.** All links are rigid, except for the diagonal link in silicone (orange in simulation, blue on the photograph).

closed configuration. We found a good agreement between our simulations and reality, with some error in the large displacements region, which we presume to be due to the flexible links exploiting the slight out-of-plane compliance in the revolute joints to deflect less.

D. Compliant Truss Gripper Finger

We design a gripper finger for a parallel-jaw gripper by augmenting a truss structure with a flexible silicone link (Fig. 7), so that the finger deforms in contact with the grasped

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2024, Yokohama, Japan. Cite as RA-L paper.

object. In this case, the rigid equivalent has zero mobility. The parallel-jaw gripper would actuate by translating the finger base perpendicularly to the object.

It has been shown that variable stiffness is advantageous for soft manipulators [30] — it is desirable to be able to grasp with both low and high stiffness. To achieve this with a single-DoF gripper, we sculpt the load-displacement curve at the end-effector so that it is relatively flat below a limiting displacement, at which the force ramps up. Note that because the rigid links form a parallelogram, the motion direction of the end-effector is always horizontal and the vertical positioning of the force application point has no influence, facilitating experimental measurements.

As for the previous example, the optimizer was able to find a link shape closely satisfying the targets. We obtain good agreement with simulation on our fabricated prototype, except for the low-force region, in which elastic forces are too low to overcome friction at the joints, thus violating static equilibrium. Note, however, that this does not interfere with the intended functionality of the finger.

E. Suspension Mechanism for Compliant Klann Leg

For any vehicle traversing uneven ground, suspension is essential. In this demonstrator, we augment the leg mechanism of a walking robot, based on the Klann linkage, with suspension. To this end, we make all links compliant except for the end-effector and base, and ask that a vertical displacement of the end-effector by a user-set obstacle size (10 mm here) corresponds to a force (5 N here) sufficient to pull up the weight of the mechanism. The rationale here is that a single leg hitting an obstacle, before the others hit the ground, should be able to lift the robot's weight.

As shown in Fig. 8, our optimizer was able to solve this problem instance. Note that, while we consider only one target point on the load-displacement curve, the problem is non-trivial as we ask for the same profile in 3 different mechanism configurations, sampled on the ground section of the leg motion cycle. This example also illustrates the benefits of decoupling forces and motions during design: a mechanism with desired motion cycle can be first designed using kinematic solvers, and our kinematics-preserving method can then solve for the force-displacement profile.

F. Performance

Our simulator and optimizer were implemented in C++. Computations were performed on a machine with an Intel Core i7-7700 processor (4 cores, 4.2 GHz) and 32 GB of RAM. Key statistics are given in Table I.

TABLE I
PERFORMANCE STATISTICS

	$\lvert\mathbf{s}\rvert$	sim. time	$\lvert\mathbf{p}\rvert$	#targets	#opt. iter.	opt. time
Claw	$77 + 16608$	49 s	69	2	39	2 h 28 min
Truss	$42 + 10188$	33 s	18	2	138	6 h 51 min
Klann	$77 + 15915$	34 s	57	3	43	9 h 21 min

$|\mathbf{s}|$: number of DoFs, given as rigid DoFs (including interfaces) + FE DoFs (excluding interfaces); *sim. time*: timing for one simulation frame when incrementally loading the optimized mechanism; $|\mathbf{p}|$: number of design parameters; *#targets*: number of force/displacement targets; *# opt. iter., opt. time*: number of optimization iterations and optimization time

VIII. CONCLUSION AND DISCUSSION

We have presented a method for the optimization of load-displacement profiles of flexible-link mechanisms, providing several demonstrations that illustrate its utility and applications. More generally, we have introduced an interface that allows for the embodied intelligence of a mechanism to be encoded via a proxy of unloaded behavior and force-displacement samples, which can then be used as an objective in gradient-based design tools. This interface could also be applicable to other design problems for soft robots, where it would allow for design problems to be tackled in isolation without having to consider the full system of the robot in its environment.

There are, however, limitations of our method, which open up avenues for future work.

So far we have only parameterized the rest shape of flexible links. While we use the degrees of freedom of mechanical joints when making adjustments to flexible links, we keep their position and orientation fixed. Parameterizing mechanical joints, and also the stiffness of the materials used for fabrication, could help to significantly increase the design space.

Our optimization currently does not consider the stress concentrations in the flexible link during deformation. By introducing additional objectives based on, e.g., the von Mises stress, our optimization could find flexible link designs with better durability.

We have focused on optimizing the design of planar mechanisms. However, most of our modeling generalizes to arbitrary spatial mechanisms, and we plan to extend our handle-based approach to optimize mechanisms with flexible links in full 3D. While the general 3D problem will likely present unforeseen challenges that are beyond the scope of this work, we expect the core building blocks to be directly applicable.

APPENDIX

We detail here the $\mathcal{C}^2$ barrier function

$$\beta(x, x_{\max}, \varepsilon, m) = \begin{cases} 0 & x \leq x_{\max} \\ m\left(\frac{x - x_{\max}}{\varepsilon}\right)^3 & x > x_{\max} \end{cases} \tag{15}$$

that we use in our regularization terms, $\mathcal{R}_{\text{mesh}}$ and $\mathcal{R}_{\text{shape}}$. x is the evaluation point, $x_{\max}$ the threshold value, and ε and m control the growth speed. For $\mathcal{R}_{\text{mesh}}$, we used $\rho_{\max} = 30$, $\varepsilon = 30$, $m = 5$.

To evaluate $\mathcal{R}_{\text{shape}}$ for a flexible link, we uniformly sample splines (500 points). At each sample, we use two soft barrier evaluations to keep the in-plane thickness δ within bounds imposed by the fabrication process. For Onyx examples, we use $\delta_{\min} = 1$ mm and $\delta_{\max} = 9$ mm. For silicone, we used $\delta_{\min} = 1.5$ mm and $\delta_{\max} = 6$ mm. In both cases, m was set to 1 and ε to 2% of the bound's value.

Finally, for the curvature term in $\mathcal{R}_{\text{shape}}$, we penalize r with β, setting $r_{\max} = 0.08$ and using $\varepsilon = 0.01$ and $m = 1$.

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2024, Yokohama, Japan. Cite as RA-L paper.

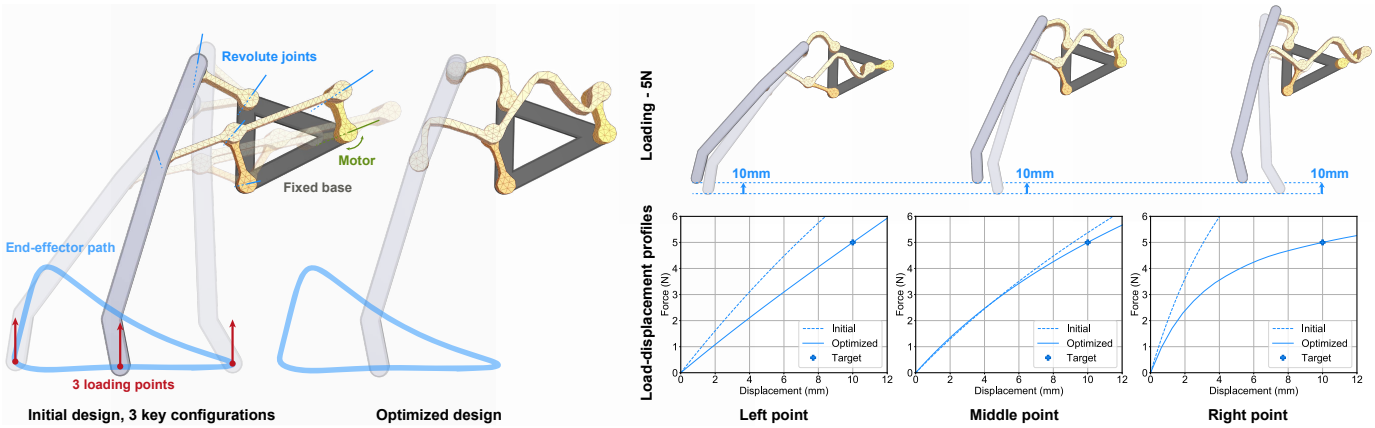

Fig. 8. **Compliant Klann leg.** All links are compliant (orange) except for the base (dark grey) and end-effector (light grey). The 3 considered configurations are shown on the left along the foot trajectory. The 3 corresponding force-displacement characteristics (initial and optimized) are shown on the right. The characteristics were computed in simulation only for this example, with the material parameters of Onyx.

REFERENCES

[1] G. Mengaldo, F. Renda, S. L. Brunton, M. Bächer, M. Calisti, C. Duriez, G. S. Chirikjian, and C. Laschi, "A concise guide to modelling the physics of embodied intelligence in soft robotics," *Nat. Rev. Phys.*, pp. 1–16, 2022.

[2] A. Jacobson, I. Baran, J. Popović, and O. Sorkine, "Bounded biharmonic weights for real-time deformation," *ACM Trans. Graph.*, vol. 30, no. 4, July 2011.

[3] A. E. Albanesi, V. D. Fachinotti, and M. Pucheta, "A review on design methods for compliant mechanisms," *Mecánica Computacional*, 2010.

[4] C. A. Mattson, "Synthesis through rigid-body replacement," *Handbook of compliant mechanisms*, pp. 109–121, 2013.

[5] L. L. Howell and A. Midha, "A Method for the Design of Compliant Mechanisms With Small-Length Flexural Pivots," *J Mech Design*, vol. 116, no. 1, pp. 280–290, Mar. 1994.

[6] J. B. Hopkins and M. L. Culpepper, "Synthesis of multi-degree of freedom, parallel flexure system concepts via freedom and constraint topology (fact) – part i: Principles," *Precision Engineering*, vol. 34, no. 2, pp. 259–270, 2010.

[7] V. Megaro, J. Zehnder, M. Bächer, S. Coros, M. Gross, and B. Thomaszewski, "A computational design tool for compliant mechanisms," *ACM Trans. Graph.*, vol. 36, no. 4, July 2017.

[8] B. Zhu, X. Zhang, H. Zhang, J. Liang, H. Zang, H. Li, and R. Wang, "Design of compliant mechanisms using continuum topology optimization: A review," *Mechanism and Machine Theory*, vol. 143, no. 103622, 2020.

[9] L. Yin and G. Ananthasuresh, "Design of distributed compliant mechanisms," *Mechanics Based Design of Structures and Machines*, vol. 31, pp. 151–179, Jan. 2003.

[10] R. H. Burns, "The kinetostatic synthesis of flexible link mechanisms," Ph.D. dissertation, Yale University, 1964.

[11] T. Shoup and C. McLarnan, "A survey of flexible link mechanisms having lower pairs," *Journal of Mechanisms*, vol. 6, no. 1, pp. 97–105, 1971.

[12] S. J. Winter and T. E. Shoup, "The displacement analysis of path-generating flexible-link mechanisms," *Mechanism and Machine Theory*, vol. 7, no. 4, pp. 443–451, 1972.

[13] N. M. Sevak and C. W. McLarnan, "Optimal Synthesis of Flexible Link Mechanisms with Large Static Deflections," *Journal of Engineering for Industry*, vol. 97, no. 2, pp. 520–526, May 1975.

[14] R. Zhang, T. Auzinger, and B. Bickel, "Computational design of planar multistable compliant structures," *ACM Trans. Graph.*, vol. 40, no. 5, Oct. 2021.

[15] P. Wang and Q. Xu, "Design and modeling of constant-force mechanisms: A survey," *Mechanism and Machine Theory*, vol. 119, pp. 1–21, 2018.

[16] N. Tolou, J. Gallego Sanchez, and J. Herder, "Statically-balanced compliant micromechanisms," *Mikroniek*, vol. 50, pp. 20–25, Jan. 2010.

[17] T. Takahashi, J. Zehnder, H. G. Okuno, S. Sugano, S. Coros, and B. Thomaszewski, "Computational design of statically balanced planar spring mechanisms," *IEEE Robot. Autom. Lett.*, vol. 4, no. 4, pp. 4438–4444, 2019.

[18] G. Radaelli, "Synthesis of mechanisms with prescribed elastic load-displacement characteristics," Ph.D. dissertation, Delft University of Technology, 2017.

[19] L. Paez, K. Melo, R. Thandiackal, and A. J. Ijspeert, "Adaptive compliant foot design for salamander robots," in *2019 2nd IEEE International Conference on Soft Robotics (RoboSoft)*, 2019, pp. 178–185.

[20] M. G. Catalano, M. J. Pollayil, G. Grioli, G. Valsecchi, H. Kolvenbach, M. Hutter, A. Bicchi, and M. Garabini, "Adaptive feet for quadrupedal walkers," *IEEE Trans. Robot.*, vol. 38, no. 1, pp. 302–316, 2022.

[21] M. Catalano, G. Grioli, E. Farnioli, A. Serio, C. Piazza, and A. Bicchi, "Adaptive synergies for the design and control of the pisa/iit softhand," *The International Journal of Robotics Research*, vol. 33, no. 5, pp. 768–782, 2014.

[22] R. Ma and A. Dollar, "Yale openhand project: Optimizing open-source hand designs for ease of fabrication and adoption," *IEEE Robot. Autom. Mag.*, vol. 24, no. 1, pp. 32–40, 2017.

[23] S. Puhlmann, J. Harris, and O. Brock, "Rbo hand 3: A platform for soft dexterous manipulation," *IEEE Trans. Robot.*, pp. 1–16, 2022.

[24] J. Nocedal and S. Wright, *Numerical optimization*. New York, NY, USA: Springer, 2006.

[25] C. Schumacher, E. Knoop, and M. Bächer, "A versatile inverse kinematics formulation for retargeting motions onto robots with kinematic loops," *IEEE Robot. Autom. Lett.*, vol. 6, no. 2, pp. 943–950, 2021.

[26] E. Sifakis and J. Barbič, "Finite element method simulation of 3d deformable solids," *Synthesis Lectures on Visual Computing*, vol. 7, pp. 1–69, Oct. 2015.

[27] J. C. Cavendish, D. A. Field, and W. H. Frey, "An approach to automatic three-dimensional finite element mesh generation," *International Journal for Numerical Methods in Engineering*, vol. 21, no. 2, pp. 329–347, 1985.

[28] V. Parthasarathy, C. Graichen, and A. Hathaway, "A comparison of tetrahedron quality measures," *Finite Elements in Analysis and Design*, vol. 15, no. 3, pp. 255–261, 1994.

[29] J. Sobieszczanski-Sobieski, J.-F. Barthelemy, and K. M. Riley, "Sensitivity of optimum solutions of problem parameters," *AIAA journal*, vol. 20, no. 9, pp. 1291–1299, 1982.

[30] H. Stuart, S. Wang, O. Khatib, and M. R. Cutkosky, "The ocean one hands: An adaptive design for robust marine manipulation," *The International Journal of Robotics Research*, vol. 36, no. 2, pp. 150–166, 2017.